Time Warping Algorithms and its Applications on Financial Time Series

by

LUI, Ga Ching Samuel

A Thesis Submitted to
The Hong Kong University of Science and Technology
in Partial Fulfillment of the Requirements for
the Degree of Master of Philosophy
in Physics

August 2017, Hong Kong

Authorization

I hereby declare that I am the sole author of the thesis.

I authorize the Hong Kong University of Science and Technology to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize the Hong Kong University of Science and Technology to reproduce the thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

LUI, Ga Ching Samuel

August 16, 2017

Time Warping Algorithms and its Applications on Financial Time Series

by

LUI, Ga Ching Samuel

This is to certify that I have examined the above Mphil thesis
and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by
the thesis examination committee have been made.

Prof. Kwok Yip Szeto, Thesis Supervisor

Prof. Michael S. Altman, Head of Department
Department of Physics

August 16, 2017

Acknowledgement

I would like to thank my supervisor Prof. Kwok Yip Szeto for his patience and guidance. I have learnt a lot from him, not just in physics but in many other things as well. I also want to thank my friends in the office, Alan, Henry, Cindy, Kelvin, Leo, Hugh and other people, who make my time here at HKUST much more colorful. It is always a joy to work with and learn from my group-mates Ho Fai Ma, Ka Wai Cheung and Degang Wu, who I have known for quite some time. Lastly, but most importantly, I am grateful for the support and love from my parents, who have always been there for me in some of the slightly more difficult situations for the past few years.

Table of Contents

List of Figures

Time Warping Algorithms and its Applications on Financial Time Series

by

LUI, Ga Ching Samuel

Department of Physics
The Hong Kong University of Science and Technology

Abstract

We introduce some of the methods for time warping, which is a technique normally used in speech recognition. Discrete time warping genetic algorithm (dTWGA) is a method based on genetic algorithm, which has been commonly used in solving optimization problems when the solution space is large and when there is no analytic form for such solution. Another method, known as dynamic time warping (DTW), makes use of dynamic programming and involves additional constraints compared to dTWGA. We illustrates the use of dTWGA on construction of financial networks. We then apply DTW on financial time series for the purpose of portfolio management. In addition to time warping techniques, we also make use of signal detection theory and concepts borrowed from fuzzy set theory in incorporating technical patterns or chart patterns used by traders and technical analysts into some objective trading strategies in a quantitative approach as contrasted to the usual practice by traders which can be seen as a subjective and qualitative approach in predicting the trend of price.

Chapter 1

Introduction

In time series analysis, There are different measurements of couplings between two time series. However, in these measurements, the non-linearity in time mapping is not dealt with. For example, correlation has been used in financial time series to construct a description of the financial network between different stocks [1, 2, 3, 4]. Stemmed from information theory, mutual information [5, 6] and transfer entropy [7, 8] is used to determine the causality or non-linear relation between two time series. However, in these measurements, the non-linearity in time mapping is not dealt with. Suppose there are two time series, $S_1(t)$ and $S_2(t)$, and suppose that $S_1(t + \Delta t) = f(S_2(t))$ for some function f, cross-correlation, transfer entropy and mutual information would not be effective if Δt is not a constant. Different methods of time warping are developed to tackle this problem.

In particular, dynamic time warping (DTW) has been proposed by Sakoe and Chiba [9], which is based on dynamic programming in recognizing spoken words. It is a pattern matching algorithm similar to the dynamic programming approach for sequence alignment. In their method, three constraints, which are the boundary conditions, step-size and monotonicity constraint, are imposed on the solutions and this allows the use of dynamic programming. Tsinaslanids et al [10] compares DTW with Pearson correlation coefficient and Spearman's correlation coefficient, and shows that DTW can be used for effective measurement of similarity between two time series. Apart from language recognition [11], DTW has been used in other areas with different applications. For instance, DTW is shown to be effective in classification of arrhythmia in ECG signals [12, 13]. To reduce the space complexity and to avoid "pathological warping" of the time series, different warping windows have been proposed [14, 15]. The complexity for the search of the optimal path in DTW without warping windows is $\mathcal{O}(N^2)$, where N is referred to here as the length of time series. And pathological warping mentioned above refers to mapping a time indice to another time indice that is far away. The warping window refers to the symmetric band around the diagonal of the matrix and only within the band is time warping allowed. In particular, genetic algorithm was used for the search in parameter space for the warping windows [16].

Genetic algorithm (GA) is a metaheuristic searching method that has been used in optimization problems. It simulates the evolution of a population of solutions to attain higher fitness, which is akin to the evolution of organisms described by Darwinism in adapting to the environment. GA has been extensively used in different disciplines, with the example of portfolio optimization [17], scheduling [18], knapsack problem [19], steller structure modelling [20], optimization of stability in financial flow network [21, 22] and reliability optimization in electric power network [23]. We have previously used GA for the purpose of time warping and named our method as TWGA, or Time Warping Genetic Algorithm. Different from DTW, TWGA only involves the boundary constraint and monotonicity constraint. However, one can always introduce the stepsize constraint back to TWGA in the form of the penalty term in the fitness function which penalizes the solutions that does not satisfy the stepsize constraint. Different from usual application of using time warping as a similarity test, we use TWGA to register the phase difference between the ECG time series captured during different apneic periods. We have also introduced a variation of the algorithm called Discrete Time Warping Genetic Algorithm and will present it in this thesis.

In existing methodologies, few have considered template-based approaches to incorporate technical patterns into automated trading schemes and time series analysis, while some of the rule-based schemes only consider the noticeable features of a pattern such as the peaks and troughs in stock charts. In this work, we apply DTW on the recognition of technical patterns in financial time series. Technical analysis has its origin in Dow Theory and is commonly used in the industry by traders and chartists. Obviously, Dow Theory does not reconcile well with random walk hypothesis and efficient market hypothesis. However, the financial market is a complex system constituted by traders, some of whom make their trading decisions based on technical analysis and would affect the price, which in turn is available globally to all traders forming a feedback mechanism. This motivates us to develop a qualitative approach towards chart patterns which are extensively used in technical analysis to capture the change of price trends. We use DTW to determine the similarity of a segment in the time series of stock price with predefined templates of chart patterns for the purpose of price prediction. This involves concepts in signal detection theory.

The Baysian Signal Detection Theory (SDT) is very important in communication engineering. The simplest example involves two hypothesis, namely the presence and the absence of a signal, and their associated actions, of which the combination gives altogether four possible outcomes [24]. When the signal is present, the action associated to the presence of the signal is classified as Hit, while the action associated to the absence of the signal is classified as Miss. On the other hand, if the signal is absent, the action associated to the presence of the signal is classified as False Alarm, and the action associated to the absence of the signal is classified as Correct Rejection. The four cases would correspond to different costs. For example, the cost of Hit should be lower than the cost of Miss since Hit refers to correct detection. Similarly, the cost of Correct Rejection should be smaller than that of False Alarm. We would use Bayes detector to minimize such cost for decision making in active portfolio management with one risky asset and cash.

Finally, we try to link up SDT with fuzzy set by Parrondo's Game. Parrondo's game was first proposed in [25] and can be seen as a simplified version of the flashing ratchet that can be used to account for the directed motion of particles [26, 27]. Optimal game sequence of Parrondo's game was studied by Dinis [28] and the effect of memory in Parrondo's Game was studied by Cheung et al [29]. Wu and Szeto [30] proposed the concept of extended Parrondo's game, which refers to the combination of two games into a better game. This inspires us to combine two strategies into a better investment strategy by introducing fuzzy set into our algorithm. Fuzzy Set was first introduced by Zedah [31] in 1965. Instead of using a crisp classification with $A \cap \neg A = \emptyset$, where A is a set, a membership function with range from 0 to 1 is used to determine the membership in A [32]. Type-2 fuzzy set is later introduced and studied [33] and it involves of the fuzzification of the membership function. An example of this is that different domain experts might give different evaluations of the membership function, and therefore modellings of the systems should also consist of the distribution of the membership function. In this work, we consider fuzzy set to account for the uncertainty in the estimation of cost in SDT.

For this thesis, we would focus on the application of time warping in portfolio management. In the first chapter, the concept of time warping would be introduced and properly formulated. We would also review genetic algorithm and dynamic programming, and explain dTWGA and DTW in details with the example of financial network and chart patterns. In the second chapter, we would have a review on Baysian signal detection theory and the estimation of likelihood ratio and threshold. The templates of the chart patterns are also given in chapter 2 and DTW was applied for portfolio management. Apart from the chart pattern templates, k-means algorithm was used for pattern discovery. Finally, we consider patterns in the time series of historic volume in combination to patterns in historic price. In the third chapter, we review the original setting of Parrondo's game and introduce a toy model that illustrates Parrondo's effect in portfolio management. We then apply the concepts of fuzzy set to SDT. The final chapter summarizes the work presented in this thesis.

Chapter 2

Time Warping Algorithms

In this chapter, we will focus on different methods for time warping. The objective of time warping will be discussed, followed by the introduction of genetic algorithm and dynamic programming in tackling general problems. Applying genetic algorithm to time warping, we introduce discrete time warping genetic algorithm (dTWGA). The logic behind is that in working on optimization problems in different fields, as few assumptions should be made as possible so as to reduce the bias introduced. The construction of financial network is then illustrated as an example of applying dTWGA. After the discussion of dTWGA, we have a brief review of dynamic time warping (DTW), which is an application of dynamic programming in time warping and requires additional constraints.

2.1 Time Warping

We have formulated the problem that is dealt with time warping in great details in our previous work. Here, we will summarize the related results and methodologies of the related papers, which are attached in the Appendix of this thesis. Essentially, we consider two time series, namely $F(t)$ and $G(t)$, such that $t \in [0, T]$. For discrete time series, we may denote the time series as $F = \{F_n = F(\frac{nT}{N+1}) | n = 0, 1, \ldots, N+1\}$ and $G = \{G_n = G(\frac{nT}{N+1}) | n = 0, 1, \ldots, N+1\}$. Before any treatments, we normalize the time series:

$$f(n) = \frac{F_n - \min(F)}{\max(F) - \min(F)}\,, \tag{2.1}$$

$$g(n) = \frac{G_n - \min(G)}{\max(G) - \min(G)}\,. \tag{2.2}$$

The objective is to obtain the non-linear mapping of the time indices $\tau = \{\tau_n = \tau(n)|n = 1, \ldots, N\}$ such that

$$\tau_n = \arg\min_{\tau'_n} \sum_{n=1}^{N} \left| f_n - g_{\tau'_n} \right| . \tag{2.3}$$

Notice that we have neglected the endpoints, namely $i = 0, N + 1$ in this formulation. This is known to be the boundary constraint, which refers to the constraint that the endpoints of f always corresponds to the endpoints of g, while the correspondence of the points between the endpoints are given by the non-linear mapping of the time indices. Notice in Eq. (2.3), we have excluded the endpoints. However, in real applications and in subsequent sections, we might also want to consider the difference of the endpoints at $n = 0, N + 1$. Suppose we can use some techniques to find the mapping such that the pairwise difference of the time series as denoted in Eq. (2.3) is minimized, having a non-zero phase shift might lead to significant changes to the magnitude of the pairwise difference due to the boundary condition if we add the extra terms of the differences at the endpoints to Eq. (2.3) as well. Obviously, such affect would be smaller if N is large, or equivalently, if the length of the time series is longer. Consider $F(t) = 2\sin(t)$ and $G(t) = \sin(t + \pi/4)$ as an example. Suppose $t \in \{0, 2\pi\}$, and we set $N = 3$ for the discrete time series. After normalization, the time series are scaled by a constant factor such that the values falls between 0 and 1. We then proceed to obtain $\tau = \{0, 1, 2\}$ by Eq. (2.3). This shows that the signals is shifted by 1 time index, corresponding to a constant phase shift of $\pi/2$ since the sampling rate of the time series is $\Delta t = T/(N + 1)$. This means that $F(t)$ lags behind $G(t)$ by a phase of $\pi/2$. Moreover, the distance given by Eq. (2.3) is zero, suggesting that the two time series are similar after time warping. However, if one include the differences at the endpoints, namely $|f_0 - g_0|$ and $|f_{N+1} + g_{N+1}|$, such distance would be one, as observed in Fig. 2.1. A method one may consider to reduce the influence of the endpoints is to apply a rolling window to one of the time series such that the difference would be of minimum when the phase shift for the sinusoidal time series is zero. We have in this example assumed the domain for both function to be the same. For functions with different domains, normalization is required along the time index, such that $T = 1$.

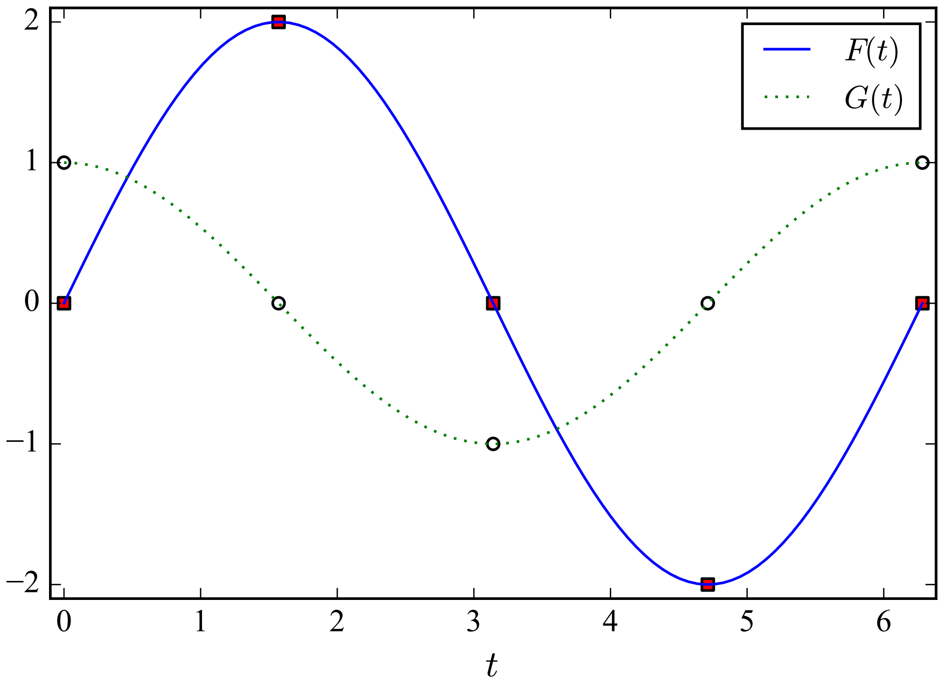


Figure 2.1: An example illustrating the concept of time warping, where the blue solid line denotes $F(t) = 2\sin(t)$ and the green dotted line denotes $G(t) = \sin(t + \pi/4)$. The filled square and the blank circles denote the discrete time series $f(n)$ and $g(n)$ respectively, such that $n = 0, 1, \ldots, 4$, with $N = 3$.

2.2 Genetic Algorithm

Before we move on to discuss dTWGA, I would give a brief introduction of genetic algorithm in this section. Genetic algorithm, or GA, is a meta-heuristic that has been widely used to solve optimization problems. Inspired by the idea of evolution and natural selection in nature, GA consists of a population of chromosomes as possible solutions to an optimization problem.

In nature, chromosomes encodes genetic information, which are expressed in the form of phenotypes. Some phenotypes are more advantageous than other phenotypes under the external pressure imposed by the environment. For example, a classical example regarding peppered moths is that the dark-colored moths have a higher chance to survive in industrial areas since they are better camouflaged compared to white moths, which are more likely to be eaten by birds. The color of the peppered moths are the phenotypes, expressing the genetic codings, or genotypes, in the chromosomes, while the external pressure is the habitat of the moths. Since the dark-colored moths have a higher chance of survival ,they are more likely to mature. As the moths reaches maturity, they are able to carry out sexual reproduction, which allows mixing and variations of genetic materials in the process of crossover. Another source of genetic variation in the gene pool is mutation, which involves sudden alterations of genetic sequences due to a number of reasons such as exposure to radiation. The genetic information is therefore able to be passed down to the offspring from their parents by reproduction with variations. Across generations, dark-colored moths become more prevalent than white moths due the the bias or asymmetry imposed by the environment. However, in the urban areas, the story is very different. Because white moths are less visible to birds in rural areas, they would slowly dominates the population over time.

GA uses similar ideas to the above example. A candidate solution of the optimization problem is encoded in a chromosome. The value of objective function in the optimization problem is dependent of the chromosome, and is defined as the fitness function. The chromosomes corresponding to higher fitness, or larger value of the fitness function, would have a higher chance to survive and be present in the next generation, whereas the chromosomes with lower fitness would have a higher probability to be removed from the population. Moreover, fitter chromosomes would also have a higher chance to be selected as parents and produce offspring, replacing the removed chromosomes that are less fit. Two operations, namely crossover and mutation, are usually involved in typical GAs. There are several crossover operators, with the examples of single-point crossover and two-point crossover. Two examples are shown in Fig. 2.2 and Fig. 2.3 showing the two operators on two parent chromosomes to produce two offspring. However, notice that we would like to keep the fittest chromosome since it is the best solution we have. Therefore, the principle of elitism is introduced such that the fittest chromosome would not be altered by the operations.

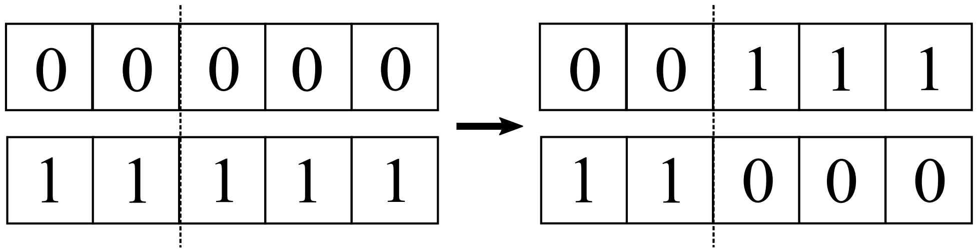


Figure 2.2: An example illustrating single-point crossover on binary string chromosomes with 5 loci, with dotted lines notating the crossover point.

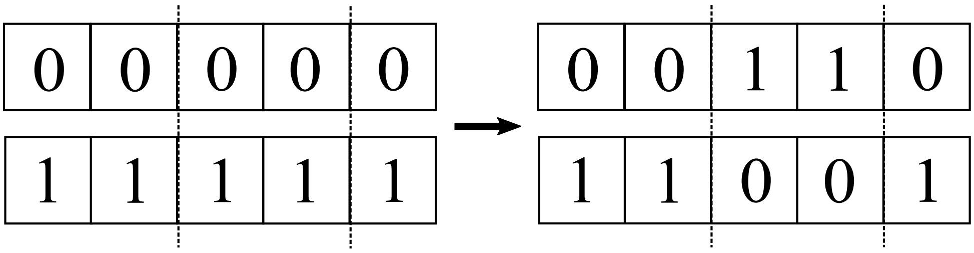


Figure 2.3: An example illustrating two-point crossover on binary string chromosomes with 5 loci, with dotted lines notating crossover points.

2.3 Discrete Time Warping Genetic Algorithm

Now that the general idea of GA is introduced, we can explore the concept in greater details with dTWGA, which is a variant of GA that does not involve crossover and can be used for time warping. This section generally follows our draft papers in the Appendix. For the purpose of time warping, two constraints are imposed: monotonicity and boundary condition. Monotonicity refers to the preserved ordering of the time index, meaning that

$$\tau_n \leq \tau_{n+1} \quad \forall n \,. \tag{2.4}$$

The second constraint, or the boundary constraint, simply refers to

$$0 \leq \tau_n \leq N+1 \quad \forall n \,. \tag{2.5}$$

Apart from these two requirements, no further constraints are imposed because we have no addition information regarding the problem concerned.

2.3.1 Encoding

Consider chromosome k, where $k = 1, 2, \ldots, N_p$ and N_p is the number of chromosomes in the population, the sequence of the chromosome can be expressed as $C_k = \{\tau'_n | n = 1, 2, \ldots, N\}$. To satisfy the boundary constraint, we assume that g_0 corresponds to f_0, whereas g_{N+1} corresponds to f_{N+1}. We therefore only consider N genes at N loci for one chromosome, such that each gene is encoded directly by the time index after warping. For the initialization of the population, $N_p \times N$ integers are drawn from discrete uniform distribution in $[1, N]$, so that each chromosome is composed of N integers. This means that each value has a $1/N$ chance to be selected. For each chromosome, the N integers are then sorted so that the monotonicity constraint is fulfilled.

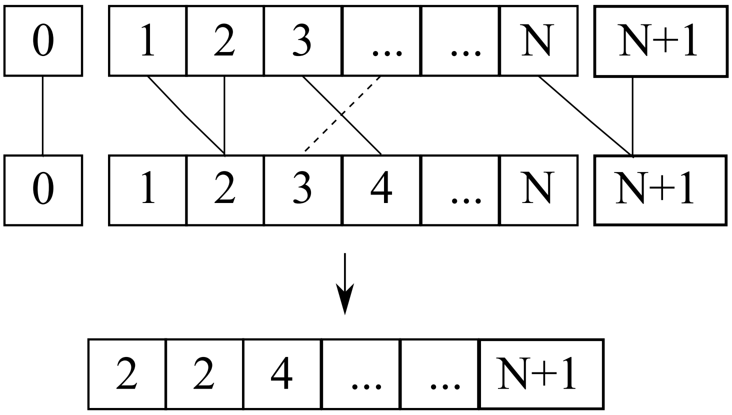


Figure 2.4: An example illustrating the time warping encoded by a chromosome with length N for dTWGA.

2.3.2 Mutation Operator

To maintain monotonicity, only mutation is involved while there is no crossover in dTWGA. In a chromosome, N_m loci are chosen randomly, at which the gene would mutate. Therefore, there are altogether $N_m \times (N_p - 1)$ mutations for each generation, and the fittest chromosome would remain unchanged. If $n = 1$, an integer would be drawn from $[0, \tau_2]$ to replace the gene at that loci. Similarly, if $n = N$, an integer would be drawn from $[\tau_{N-1}, N + 1]$ to be the new value of τ_N, so that boundary condition is satisfied. For other values of n, an integer is selected randomly from $[\tau_{n-1}, \tau_{n+1}]$ to replace the original gene. This indicates that we are adopting a mutation probability of N_m/N, which is a variable that would be more convenient to use in extensions of the mutation operator to chromosomes with different lengths.

2.3.3 Fitness Function

In dTWGA, chromosomes evolve to attain a smaller distance between the time series after time warping. We define this distance to be cost function s_k for chromosome k, or C_k:

$$s_k = |f_0 - g_0| + |f_{N+1} - g_{N+1}| + \sum_{n=1}^{N} |f_n - g_{\tau_n}| \tag{2.6}$$

Because this is a minimization problem, the chromosome with a lower cost, or smaller value of s_k, would be referred to as a fitter chromosome. On the contrary, the chromosome with large s_k is said to have a low fitness. The chromosomes are then sorted so that

$$s_k \leq s_{k+1} \quad \forall k\,. \tag{2.7}$$

Typically, GAs make use of different methods for selection such as Russian Roulette, for which the probability of parent selection for each chromosome is dependent of the fitness. The ones with a higher fitness would have a larger probability to be selected as a parent. In dTWGA, the chromosome with a larger cost would have a smaller chance to be selected, which is the opposite of maximization problems.

2.3.4 Selection

Suppose we would like to remove N_k chromosomes from the population with the largest values of k and replace them with ones selected from the fitter $N_p - N_k$ chromosomes. For the method of Russian Roulette, a typical choice of selection probability for parents in solving maximization problems is

$$P_p(k) = \frac{s_k}{\sum\limits_{j=1}^{N_p-N_k} s_j} . \tag{2.8}$$

For minimization, one may consider using

$$P_p(k) \propto 1 - \frac{s_k}{\sum\limits_{j=1}^{N_p-N_k} s_j} . \tag{2.9}$$

Since Eq. (2.9) has to be summed to unity, for each replacement of chromosomes, we define the probability for each chromosome among the fittest $N_p - N_k$ chromosomes to be selected in a bootstrap manner as

$$P_p(k) = \frac{\sum\limits_{\substack{i=1 \\ i\neq k}}^{N_p-N_k} s_i}{(N_p - N_k - 1) \sum\limits_{j=1}^{N_p-N_k} s_j} . \tag{2.10}$$

The denominator is the normalization constant, so that P_p is not larger than 1. Notice that P_p not only depends on fitness of individual chromosome, but also depends on the population chromosome, which is important for the convergence of solution. Therefore, the chromosome with a smaller s_k would have a larger probability to be selecting as a parent when replacing the weaker chromosomes.

2.3.5 Iteration

The population is initialized randomly as described above, which is the 0-th generation, or $g = 0$. Then, the process of obtaining the value of cost function, sorting through a population of chromosomes is assign new values of k, parent selection to remove weaker chromosomes and gene mutation are iterated until for $g = G$, where G is some large integer. In other words, we use the number of generations as the stopping criteria. The entire algorithm is illustrated in Fig. 2.5, and the fittest chromosome, C_1, at $g = G$ corresponds to the time mapping between the two functions, with difference s_1.

$$d(f, g) = s_1 \tag{2.11}$$

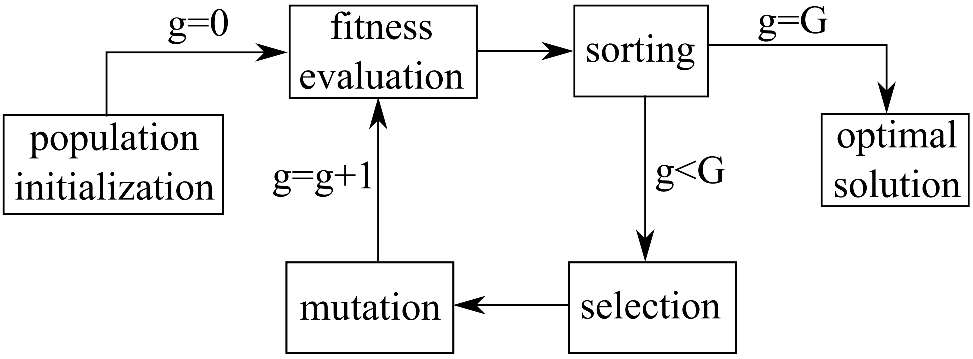


Figure 2.5: Flow chart of dTWGA.

2.4 Example: Financial Network

In this section, we demonstrate the use of dTWGA using an example of financial network. Suppose we are considering S_i, which is the logarithmic return of the daily closing price P_i of stock i:

$$S_i(t) = \ln P_i(t) - \ln P_i(t-1) \tag{2.12}$$

We may consider N_s stocks, such that $i = 1, 2, \ldots, N_s$. Each node in the network is regarded as one of the N_s stocks, and there are altogether N_s nodes in the network. To determine the coupling between two nodes, we consider the entry in the adjacency matrix to be

$$d_{ij} = \min(d(s_i, s_j), d(s_i, -s_j)) \,, \tag{2.13}$$

This measurement is similar to the one suggested by [10] as an alternative in evaluating correlations, and $d(s_i, -s_j)$ is to account for negative correlation. The negative sign in $-s_j$ refers to flipping the time series before normalization. Smaller d_{ij} therefore implies stronger coupling between node i and node j. Considering only undirected network, we only need to consider $N_s(N_s-1)/2$ entries, while the diagonal elements are all zeros. In fact, no time warping is required to minimize the distance for the diagonal elements since the distance of a time series from itself is zero. The advantage of this construction of financial network using time warping rather than using Pearson correlation or mutual information is that the time mapping is non-linear. For example, using cross-correlation requires determining the value of time shift τ which is assumed to be constant. However, such time shift may depend on a number of factors and dTWGA is capable of dealing with time shifts that are not constant. And for this purpose, TWGA was used to determine the phase shift between two quasiperiods of the ECG signal in our previous work in the Appendix. The minimum spanning tree of the financial network at a particular instance among all 30 components of Dow Jones Index is shown in Fig. 2.6.

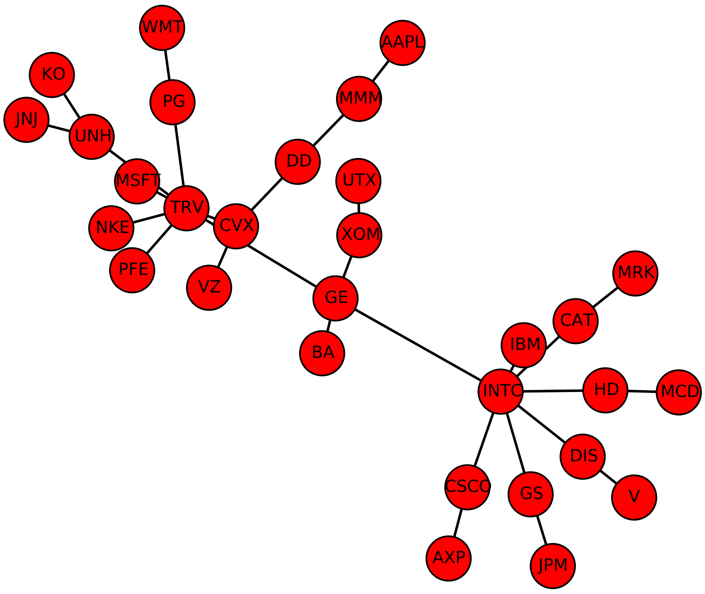


Figure 2.6: An example of a Minimum Spanning Tree of Stock Network with nodes representing the components of Dow Jones Index and weights of edges representing the distance between a pair of stocks obtained with dTWGA.

2.5 Dynamic Programming

Dynamic program (DP) is proposed by Bellman [34, 35] to avoid solving optimization problems involving sequential operations with brute force by enumerating all possible paths or cases. In DP, the Principle of Optimality, which states that "An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision" [36] was applied and the optimization problem can therefore be formulated in terms of functional, which is known as Bellman's equations. This principle is similar to solving extensive-form games with perfect information in game theory, which concerns with players making decisions sequentially. The first player would consider which of the optimized strategies of the second player would lead to a better payoff when making his or her move by backward induction, and subgame perfect Nash equilibrium can be reached assuming rationality of the players. In other words, DP involves formulating the problem in an iterative manner, and is applicable to optimization problems with optimal substructure and overlapping sub-problems , which allows us to solve such problems by solving sub-problems recursively in a bottom-up manner to obtain the value of the optimized parameter [37]. With optimal substructure, the optimal solution of an optimization problem is formed by optimal solutions of the sub-problems. On the other hand, overlapping sub-problems refers to repeat appearance of a few sub-problems in solving a larger optimization problem. The solution that leads to such value can then be traced back. The execution of DP is illustrated with DTW in the next section.

2.6 Dynamic Time Warping

DTW is based on dynamic programming and has been used extensively in speech recognition. It consists of constructing a cost matrix function $\mathbf{C}$, such that

$$C_{nn'} = (f_n - g_{n'})^2 \, . \tag{2.14}$$

The objective of DTW is to find the optimal path $P_\star$ from all possible paths $P_k = \{P_k^0 = (0,0), \ldots, P_k^i = (n, n'), P_k^I \ldots, (N+1, N+1)\}$, where P_k^i refers to the i-th step for path k and I denoting the length of the path, so that the sum of distance is minimized

$$d(f, g) = \sum_{(n,n')\in P_\star} C_{nn'} \, . \tag{2.15}$$

Notice that the first entry of P has to be $(0,0)$ while the final entry of the path has to be $(N+1, N+1)$ in order to satisfy the boundary condition. In addition to the monotonicity constraint in Eq. (2.4) and the boundary condition in Eq. (2.5), DTW has imposed an additional constraint to limit the step size:

$$P_k^{i+1} - P_k^i \in \{(1,1),(0,1),(1,0)\}\,. \tag{2.16}$$

Now we can break down this problem into smaller sub-problems and solve it recursively by constructing the accumulative cost matrix

$$D_{nn'} = C_{nn'} + \min(D_{n-1,n'-1}, D_{n,n'-1}, D_{n-1,n'}) \tag{2.17}$$

Hence, the distance between the two time series can be obtained with $d(f,g) = D_{N+1,N+1}$. As for the time mapping, the path can be obtained by tracing back from $D_{N+1,N+1}$ with

$$P_\star^{i-1} = \begin{cases} (0, n'-1) & \text{if } n = 0 \\ (n-1, 0) & \text{if } n' = 0 \\ \arg\min(D_{n-1,n'-1}, D_{n,n'-1}, D_{n-1,n'}) & \text{otherwise} \end{cases} \tag{2.18}$$

Similar to previous sections, we have assumed that the cost matrix is a square matrix. Consider the two normalized time series that are plotted in Fig. 2.7. The cost matrix and accumulative cost matrix are plotted in Fig. 2.9, and Fig. 2.10 respectively. The time series after time warping with DTW are plotted in Fig. 2.8, which minimizes the area difference under the lines.

The major advantage of DTW over dTWGA is that the solution found by DTW is globally optimal given the constraints. For dTWGA, the time warping encoded in the best chromosome is only optimal is the local sense, and is dependent of the initialization of the chromosome as well as the mutation operations, which involves random selections. However, the advantage of using dTWGA is that it requires less constraints. More specifically, step-size constraint is not imposed. Furthermore, dTWGA uses the number of generations G as the stopping criteria. The selection of G depends on the convergence rate. The time required for computation is proportional to the population size and length of the time series for mutation. On the other hand, since DTW involves computing the accumulative cost matrix, which is an $N \times N$ matrix, the time complexity and space complexity are both $\mathcal{O}(N^2)$. Therefore, it would be computationally expensive to apply DTW on long time series.

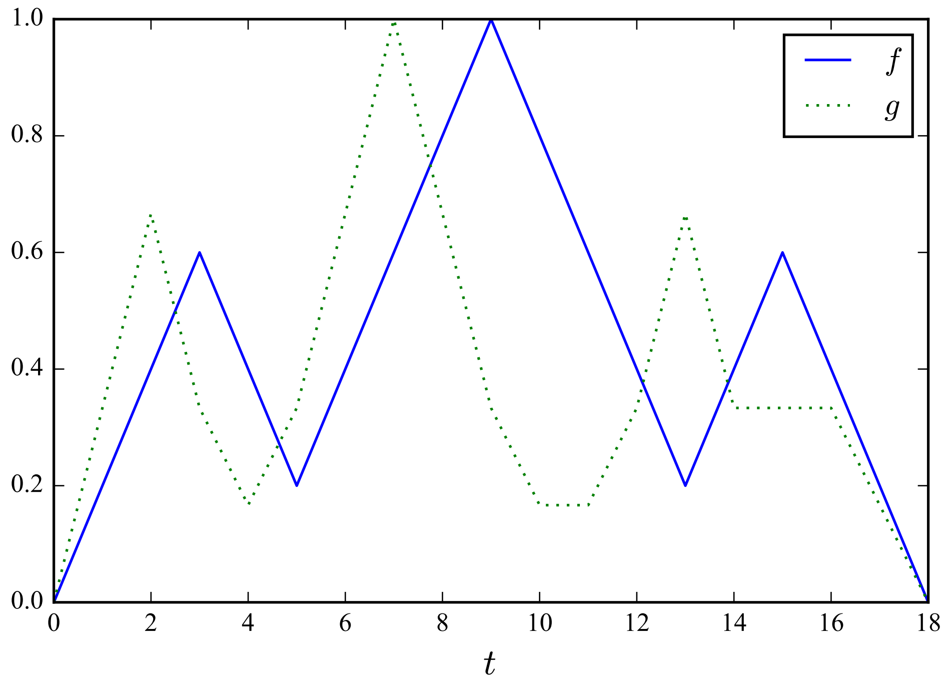


Figure 2.7: Two normalized time series $f(t)$ and $g(t')$ are plotted without time warping, which would give a larger area difference compared to that obtained with DTW.

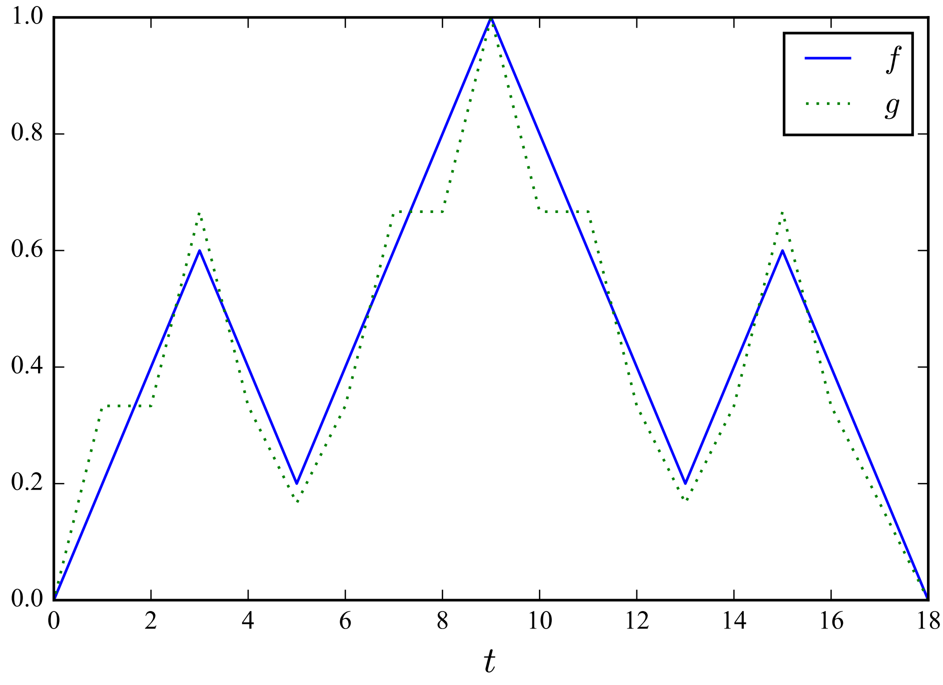


Figure 2.8: An example illustrating the use of DTW, with the time series $f(t)$ and $g(\tau(t'))$ after time warping with DTW plotted for comparison, which suggests that f and g are similar.

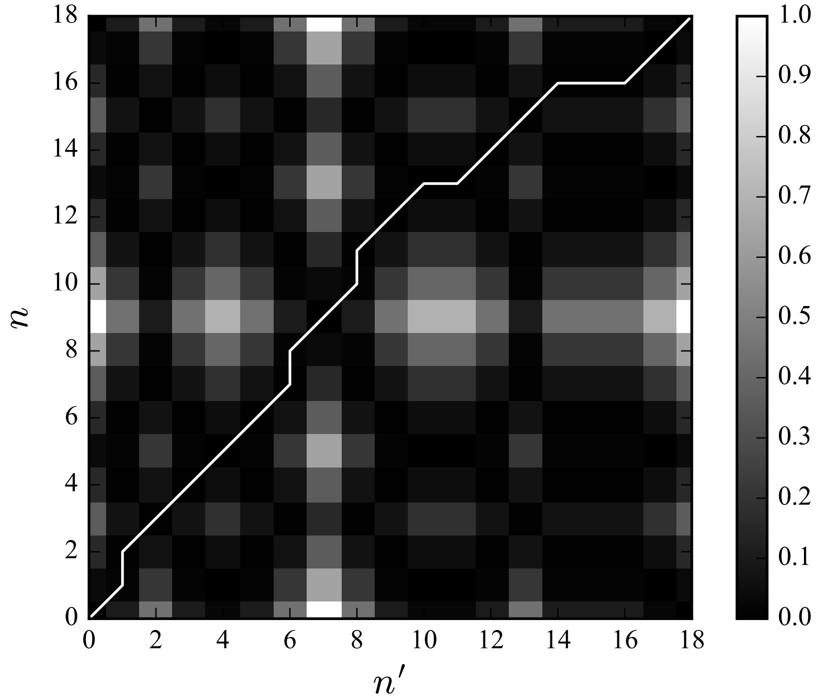


Figure 2.9: An example of cost matrix obtained in DTW, with white-colored path showing the path with minimum distance.

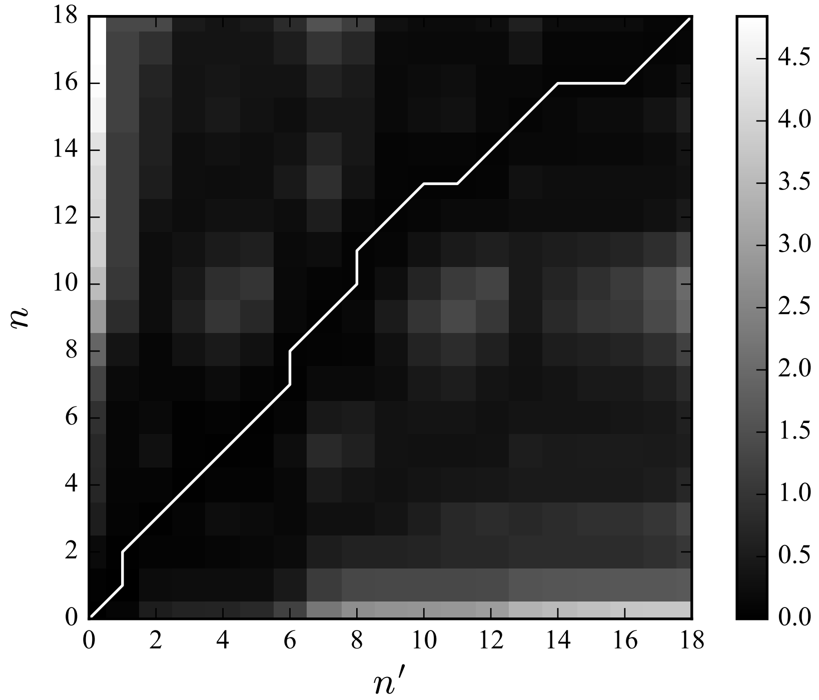


Figure 2.10: An example of accumulative cost matrix obtained in DTW, with white-colored path showing the path with minimum distance.

2.7 Summary

In this chapter, we have introduced the concept and the objective of time warping on time series. We then discussed two approaches for time warping. The first approach is based on genetic algorithm, and is coined with the name Discrete Time Warping Genetic Algorithm. We demonstrate one of the possible applications of dTWGA, which is to obtain the adjacency matrix of a financial network. The second approach, which is based on dynamic programming, is called dynamic time warping. We will adopt DTW as our main approach in later sections.

Chapter 3

Portfolio Management

In this chapter, main concepts of Bayesian decision theory will be briefly introduced. Using the methodology of time warping discussed in the previous chapter, we would quantitatively apply some of the ideas borrowed from technical analysis in finance on managing portfolio with one risky asset and cash.

3.1 Bayesian Signal Detection

3.1.1 Bayes Detector

Suppose we are dealing with a K-class classification problem, we may define λ_{ik} to be the cost incurred when an input signal $\mathbf{x}$ is assigned to support hypothesis $\mathcal{H}_i$ while the actual case is $\mathcal{H}_k$, where $i, k = 0, 1, ..., K-1$. Then, the expected risk $\mathcal{R}$ of such decision α_i of the assignment is

$$\mathcal{R}(\alpha_i|\mathbf{x}) = \sum_{k=0}^{K-1} \lambda_{ik} P(\mathcal{H}_k|\mathbf{x}) \,, \tag{3.1}$$

where $P(a|b)$ denotes the conditional probability of a given b. The optimal decision rule is to adopt α_i for a particular i such that the expected risk is minimized. For K=2, we can further define the discriminant function to be

$$g(\mathbf{x}) = \mathcal{R}(\alpha_1|\mathbf{x}) - \mathcal{R}(\alpha_0|\mathbf{x}) \,. \tag{3.2}$$

If $g(\mathbf{x}) > 0$, the input signal is classified as $\mathcal{H}_0$:

$$\lambda_{10} P(\mathcal{H}_0|\mathbf{x}) + \lambda_{11} P(\mathcal{H}_1|\mathbf{x}) > \lambda_{00} P(\mathcal{H}_0|\mathbf{x}) + \lambda_{01} P(\mathcal{H}_1|\mathbf{x}) \tag{3.3}$$

$$\eta \equiv \frac{P(\mathcal{H}_0)(\lambda_{10} - \lambda_{00})}{P(\mathcal{H}_1)(\lambda_{00} - \lambda_{11})} > \frac{p(\mathbf{x}|\mathcal{H}_1)}{p(\mathbf{x}|\mathcal{H}_0)} \equiv \Lambda(\mathbf{x}) \,, \tag{3.4}$$

where η is the threshold and Λ is the likelihood ratio. Otherwise, $\mathcal{H}_1$ is chosen. The last line in Eq. (3.3) is obtained from Bayes' rule:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}\,. \tag{3.5}$$

Therefore, to compute the likelihood ratio and threshold, we simply need to obtain the likelihood $p(\mathbf{x}|C)$ and the prior $P(C)$ instead of having to deal with the posterior $P(C|\mathbf{x})$.

3.1.2 Probability Density Estimation

For the purpose of portfolio management, we consider the rate of return of a financial time series

$$x(t) = \frac{s(t) - s(t-\Delta t)}{s(t-\Delta t)}\,. \tag{3.6}$$

If $x \geq x_d$ for a fixed value of x_d, the return for that day is classified as case "UP", denoted by the letter "U". However, if $x \leq -x_d$, the return for that day is classified as case "DOWN", denoted by the letter "D". Notice that for non-zero x_d, $P(U) + P(D) \leq 1$. If the prediction of the return at time t is U, the investor will buy the stocks; if the prediction of the return at time t is D, the investor will sell the stocks. For the likelihood, there are different ways to approximate the probability density functions. Here, we consider using sigmoid function to approximate the cumulative distribution function:

$$F_s(x) = \frac{1}{1 - e^{-\beta(x-x^\star)}}\,. \tag{3.7}$$

The value of $x^\star$ can be found at $F(x^\star) = 0.5$, where F is the empirical cumulative distribution function. Taking derivative of the sigmoid function with respect to x gives

$$\hat{p} = \frac{dF_s}{dx} = \beta F_s(x)(1 - F_s(x))\,. \tag{3.8}$$

Therefore, the value of beta can be approximated to be

$$\beta = 4F'(x^\star)\,. \tag{3.9}$$

By taking derivative of F_s with respect to x, we can obtain estimator of the probability density function, denoted with $\hat{p}$, for the evaluation of the likelihood ratio.

3.2 Chart Patterns

In technical analysis, there are different types of price patterns that signals trend continuation or reversal [38, 39, 40]. In this work, we would look at some of the common and major chart patterns used by investors.

3.2.1 Trend Reversal Patterns

The chart patterns head-and-shoulders, inverted head-and-shoulder pattern, rounded tops, rounded bottoms, double tops and double bottoms belong to the class of trend reversal patterns which signals the reversal of the current trend. The head-and-shoulder function (HS) is defined below in Eq. (3.10) and inverted head-and-shoulder is defined to be $IHS = 1 - HS$.

$$HS = \begin{cases} -\frac{4x}{2.5} + 4 & x \leq 2.5 \\ \frac{6x}{2.5} - 6 & 2.5 \leq x \leq 5 \\ -\frac{6x}{2.5} + 18 & 5 \leq x \leq 7.5 \\ \frac{4x}{2.5} - 12 & 7.5 \leq x \leq 10 \end{cases}, \quad IHS = \begin{cases} \frac{4x}{2.5} - 3 & x \leq 2.5 \\ -\frac{6x}{2.5} + 7 & 2.5 \leq x \leq 5 \\ \frac{6x}{2.5} - 17 & 5 \leq x \leq 7.5 \\ -\frac{4x}{2.5} + 13 & 7.5 \leq x \leq 10 \end{cases} \tag{3.10}$$

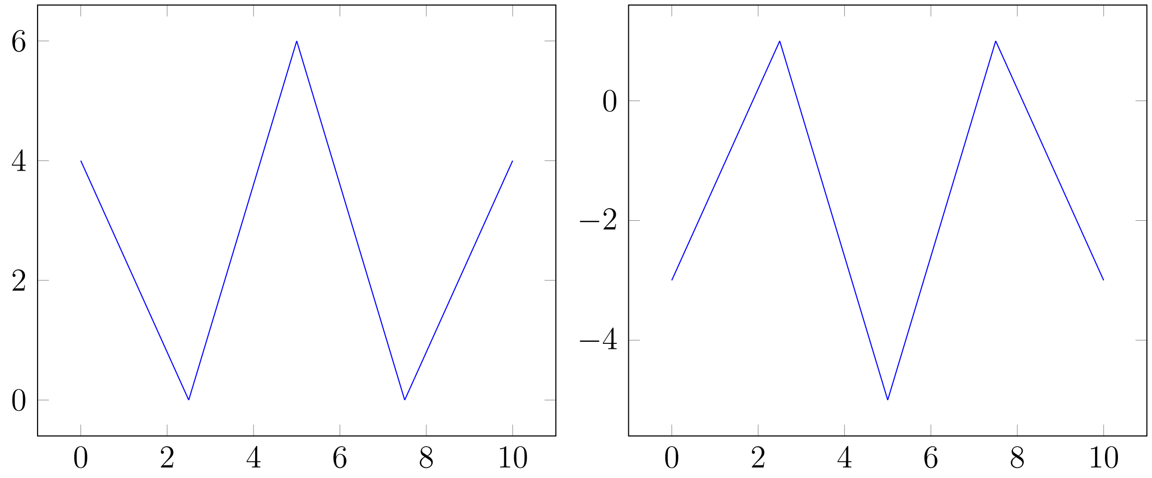


Figure 3.1: Head-and-shoulder pattern (left) and inverted head-and-shoulder pattern (right).

Rounded-top (RT) and rounded-bottom (RB) are defined in the equation below, and notice that rounded bottom is defined as $RB = 1 - RT$ in Eq. (3.11). Similarly, the pair of double-top (DT) and double-bottom (DB) are also defined in Eq. (3.12).

$$RT = \begin{cases} 0.6x & x \leq 1.5 \\ \frac{x}{35} + \frac{6}{7} & 1.5 \leq x \leq 5 \\ -\frac{x}{35} + \frac{8}{7} & 5 \leq x \leq 8.5 \\ -0.6x + 6 & 8.5 \leq x \leq 10 \end{cases}, \quad RB = \begin{cases} -0.6x + 1 & x \leq 1.5 \\ -\frac{x}{35} + \frac{1}{7} & 1.5 \leq x \leq 5 \\ \frac{x}{35} - \frac{1}{7} & 5 \leq x \leq 8.5 \\ 0.6x - 5 & 8.5 \leq x \leq 10 \end{cases} \tag{3.11}$$

$$DT = \begin{cases} \frac{x}{2.5} & x \leq 2.5 \\ -\frac{0.4x}{2.5} + 1.4 & 2.5 \leq x \leq 5 \\ \frac{0.4x}{2.5} - 0.2 & 5 \leq x \leq 7.5 \\ -\frac{x}{2.5} + 4 & 7.5 \leq x \leq 10 \end{cases}, \quad DB = \begin{cases} -\frac{x}{2.5} + 1 & x \leq 2.5 \\ \frac{0.4x}{2.5} - 0.4 & 2.5 \leq x \leq 5 \\ -\frac{0.4x}{2.5} + 1.3 & 5 \leq x \leq 7.5 \\ \frac{x}{2.5} - 3 & 7.5 \leq x \leq 10 \end{cases} \tag{3.12}$$

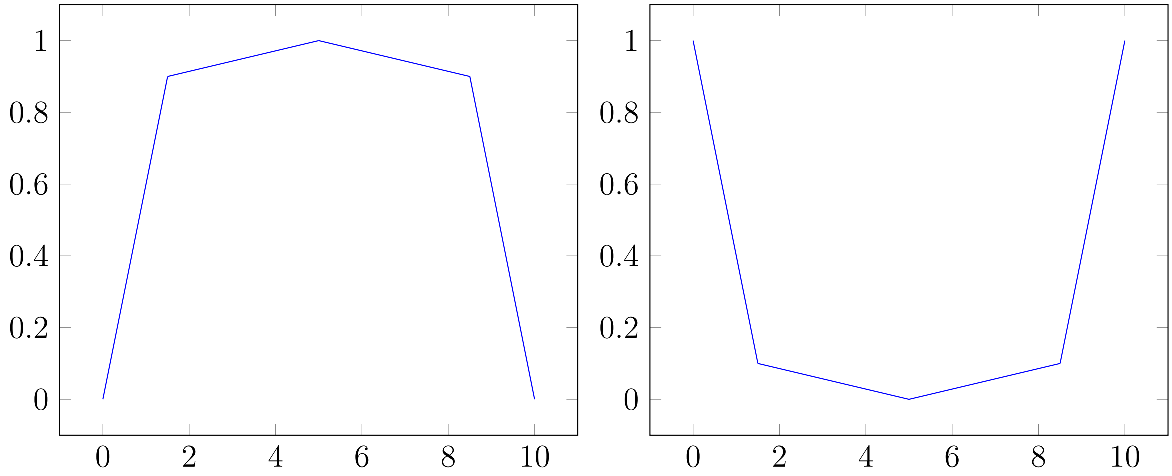


Figure 3.2: Rounded-top pattern (left) and rounded-bottom pattern (right).

3.2.2 Continuation Patterns

Another class of patterns are the continuation patterns, which includes descending broadening-top, ascending broadening-top,ascending broadening-bottom, descending broadening-bottom. We define the descending broadening-top (BTOPD) using the equation below, while the ascending broadening-bottom (BBOTU) is defined to be $BBOTU = 1 - BTOPD$ in Eq. (3.13).

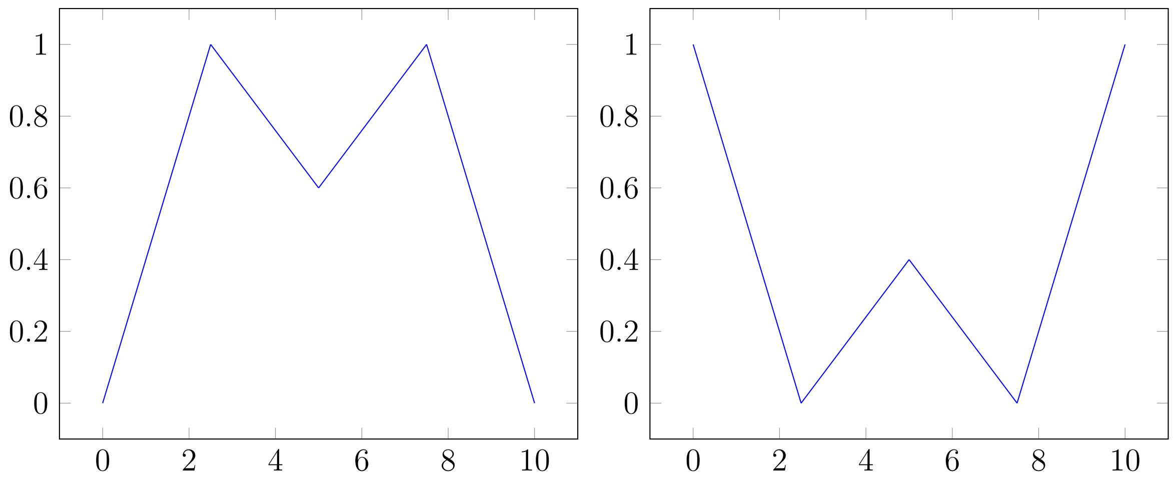


Figure 3.3: Double-top pattern (left) and double-bottom pattern (right).

$$BTOPD = \begin{cases} -\frac{0.6x}{2.5} + 1 & x \leq 2.5 \\ \frac{0.6x}{2.5} - 0.2 & 2.5 \leq x \leq 5 \\ -\frac{x}{2.5} + 3 & 5 \leq x \leq 7.5 \\ \frac{x}{2.5} - 3 & 7.5 \leq x \leq 10 \end{cases}, \quad BBOTU = \begin{cases} \frac{0.6x}{2.5} & x \leq 2.5 \\ -\frac{0.6x}{2.5} + 1.2 & 2.5 \leq x \leq 5 \\ \frac{x}{2.5} - 2 & 5 \leq x \leq 7.5 \\ -\frac{x}{2.5} + 4 & 7.5 \leq x \leq 10 \end{cases} \tag{3.13}$$

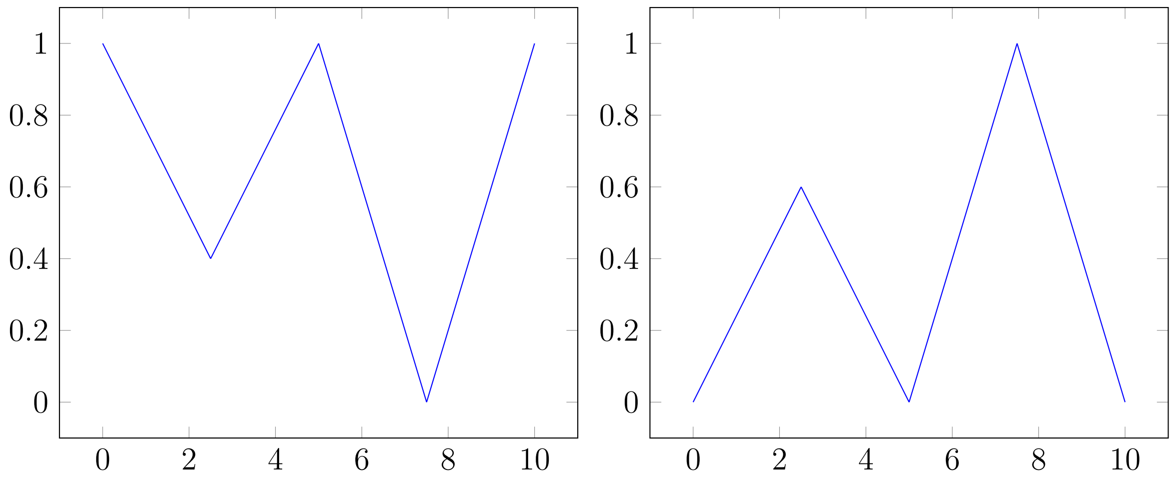


Figure 3.4: Descending broadening-top pattern (left) and ascending broadening-bottom pattern (right).

Similarly descending broadening-bottom (BBOTD) is defined below, while ascending broadening-top is simply $BTOPU = 1 - BBOTD$ in Eq. (3.14).

$$BBOTD = \begin{cases} \frac{0.4x}{2.5} + 0.6 & x \leq 2.5 \\ -\frac{0.6x}{2.5} + 1.6 & 2.5 \leq x \leq 5 \\ \frac{0.6x}{2.5} - 0.8 & 5 \leq x \leq 7.5 \\ -\frac{x}{2.5} + 4 & 7.5 \leq x \leq 10 \end{cases}, \quad BTOPU = \begin{cases} -\frac{0.4x}{2.5} + 0.4 & x \leq 2.5 \\ \frac{0.6x}{2.5} - 0.6 & 2.5 \leq x \leq 5 \\ -\frac{0.6x}{2.5} + 1.8 & 5 \leq x \leq 7.5 \\ \frac{x}{2.5} - 3 & 7.5 \leq x \leq 10 \end{cases} \tag{3.14}$$

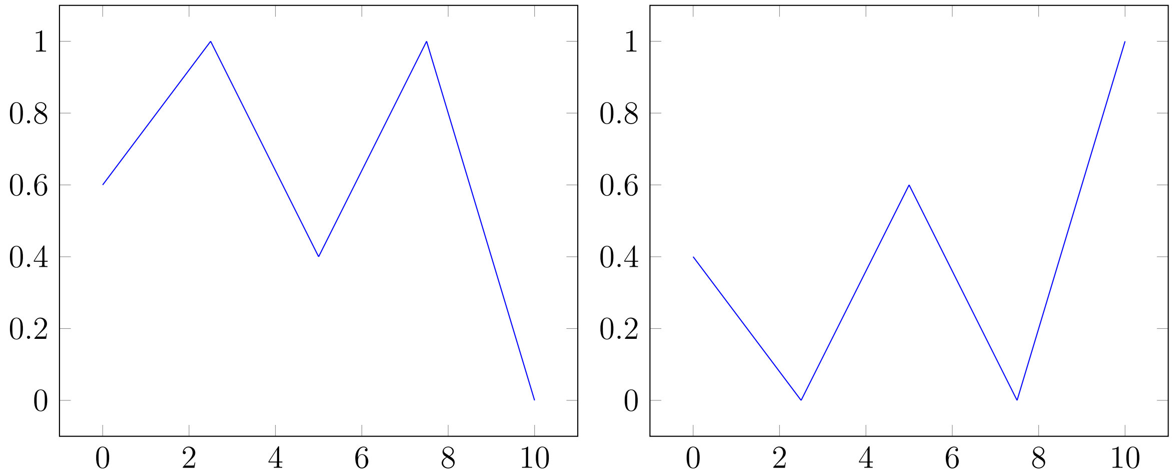


Figure 3.5: Descending broadening-bottom pattern (left) and ascending broadening-top pattern (right).

All the four examples above are broadening patterns tilted either in the upward direction or downward direction.

3.2.3 Others

Finally, in addition to these common patterns, we would also like to introduce or define two patterns which are not technical pattern yet very intuitive: straight-up (SU) and straight-down (SD).

$$SU = x; \quad SD = -x \tag{3.15}$$

3.3 Model Training

For training set $\{x^{(\ell)}\}_{\ell=1}^{N'}$ of $N' = N - \Delta t$ records of return of the stock prices for time unit Δt , the likelihood ratio as a function of $x(t')$ or $x^{(\ell)}$ is obtained by taking $\mathcal{H}_0$ to be

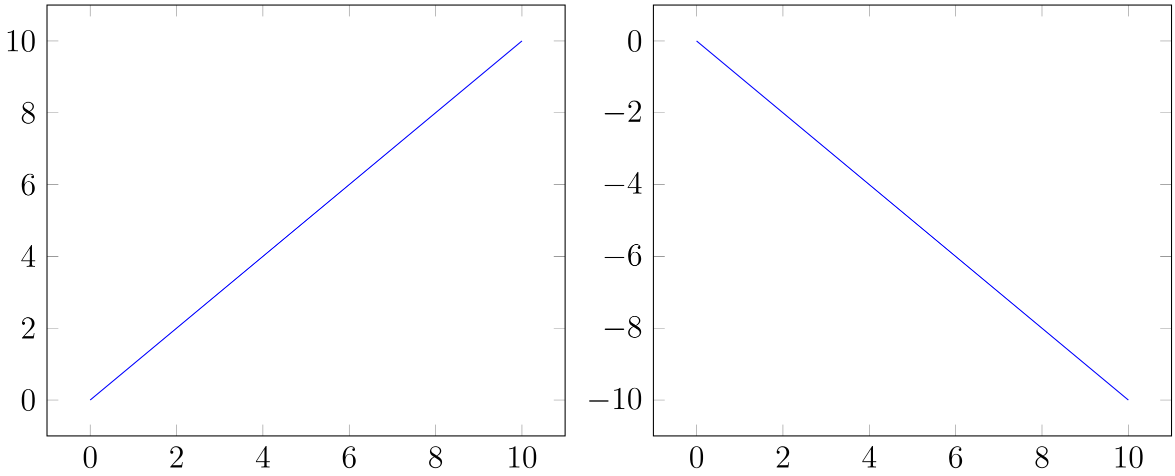


Figure 3.6: Straight-up pattern (left) and straight-down pattern(right).

U and $\mathcal{H}_1$ to be D at $t = t' + \Delta t$. $P(x|\mathcal{H}_0)$ and $P(x|\mathcal{H}_1)$ are obtained using the procedure described in Sect. 3.1.2. We may proceed to define

$$S_0 = \{x(t')|x(t' + \Delta t) \geq x_d\}, \quad S_1 = \{x(t')|x(t' + \Delta t) < -x_d\} \tag{3.16}$$

To obtain the threshold, the cost of decision, namely λ, has to be evaluated. For the evaluation of the expected cost, we first consider the probability of $\mathcal{H}_0$ and $\mathcal{H}_1$.

$$p_k = \frac{|S_k|}{|S_0| + |S_1|}\,, \tag{3.17}$$

where $k = 0, 1$ for U and for D respectively. Since it is not practical for individual investors to execute frequent trading due to transaction fee, we only consider trading when certain predetermined rules are satisfied. Hence, the estimated costs should reflect certain information when these rules are satisfied. Suppose there are two rules, namely C_0 which concerns with the trading when U is predicted by the Bayes detector, and C_1 which concerns with the trading when D is predicted, and we use C_i to denote rule i in general, then we can obtain the following set from the training set:

$$R_i = \{x(t' + \Delta t)|C_i \text{ is satisfied at } t'\}\,. \tag{3.18}$$

We further define the average return when C_0 and C_1 are satisfied:

$$m_k = \frac{1}{|R_k|}\sum_{x^{(\ell)} \in R_k} x^{(\ell)}\,. \tag{3.19}$$

To evaluate the rate of correct prediction or classification, we define the following quantity:

$$P_{ck} = \frac{1}{|R_k|}\sum_{x^{(\ell)} \in R_k} \left[(\alpha^{(\ell)} - k)^2 - \gamma^{(\ell)} k\right]\,, \tag{3.20}$$

where α and γ are defined for succinct expression for the two cases of C_0 and C_1,

$$\alpha^{(\ell)} = \begin{cases} 1 & \text{if } x^{(\ell)} \geq x_d \\ 0 & \text{else} \end{cases} ; \quad \gamma^{(\ell)} = \begin{cases} 1 & \text{if } -x_d < x \leq x_d \\ 0 & \text{else} \end{cases} . \tag{3.21}$$

Therefore, the threshold can be evaluated using the following expression:

$$\eta(R_0, R_1) = \frac{|m_o P_{c0}| p_o}{|m_1 P_{c1}| p_1} . \tag{3.22}$$

We consider the technical patterns in defined in Sect. 3.2 to be the rules. More specifically, we select two out of twelve patterns, namely Π_0 and Π_1, for rules C_0 and C_1 respectively such that corresponding actions $\alpha_0(\alpha_1)$ for $U(D)$ will be taken according to the inequality given by Bayesian signal detection theory only if C_0 (C_1) is satisfied. We denote pattern i as $\hat{y}_i$, for which the definitions are given in Sect. 3.2. To determine if C_0 or C_1 is satisfied, we first define another variable w, which is the window size of the pattern. Our objective is to observe or determine if a particular segment of the time series $s_{t'} : \{s(t'-w+1) : s(t')\}$ is similar to the templates of the chart patterns defined previously. For fair comparison, we sample w points uniformly from each pattern, of which the function's domain is [0,10]: The sampled pattern $h_i^s = \{h_{i1}^s = \hat{y}_i(0), \ldots, h_{iw}^s = \hat{y}_i(10)\}$ is then normalized to obtain $h_i = \{h_{i1}, \ldots, h_{iw}\}$ so that the range lies in $[0,1]$.

$$h_{ij} = \frac{h_{ij}^s - \min\{h_i^s\}}{\max\{h_i^s\} - \min\{h_i^s\}} . \tag{3.23}$$

We would select two different patterns for the rules, namely $\Pi_0 = h_i$ and $\Pi_1 = h_{i'}$ for some i and i'. The condition is satisfied at $t = t'$ if the following relation is true:

$$C_k : D(\Pi_k, s_{t'}) \leq d , \tag{3.24}$$

where $D(a(t), b(t))$ is the distance obtained by dynamic time warping between time series $a(t)$ and $b(t)$, while d is the sensitivity. The notation $s_{t'}$ here represents the stock price. Notice that while C_0 and C_1 are assumed to be mutually exclusive, that might not be the case according to the above prescribed procedures. However, for this section, C_0 would be evaluated first and if C_0 is satisfied at $t = t'$, we would not proceed to calculate the distance between $s_{t'}$ and h_1 and assume C_1 is not satisfied for simplicity. Our choice involves the bias or preference towards U predictions. The justification is that p_o is usually larger than p_1 in stock price. One may also consider other choices or simplifications that does no involve such bias, but we will not discuss them here. Furthermore, if C_k is satisfied at $t = t'$, then C_k would not be evaluated for $t' < t \leq (t' + w/2)$. This is to avoid the problem that a single event of matched pattern is identified across several instances, which is possible as time warping allows non-linear mapping between the time indices. Potentially, some errors might be incurred by this simplification due to the dependence of distance measurement on phase difference as discussed in Sect. 2. The justification is that although it is possible to have a smaller value of distance D between $t = t'$ and $t = t' + w/2$, we do not have enough information at $t = t'$ determine whether D would increase or decrease. These simplifications allow the training to be less computationally demanding.

To determine which patterns h_i to use as Π_0 and Π_1, we would compare a particular pattern with the time series across a rolling window. In layman's term, a rolling window refers to observing segments of the time series with fixed length at different instances. We then evaluate the following quantity for each pattern:

$$e_i(d) = \frac{\left[\sum\limits_{x^{(\ell)} \in R_i} (\alpha^{(\ell)})^2\right] - \left\{\sum\limits_{x^{(\ell)} \in R_i} \left[(\alpha^{(\ell)} - 1)^2 - \gamma^{(\ell)}\right]\right\}}{\left[\sum\limits_{x^{(\ell)} \in R_i} (\alpha^{(\ell)})^2\right] + \left\{\sum\limits_{x^{(\ell)} \in R_i} \left[(\alpha^{(\ell)} - 1)^2 - \gamma^{(\ell)}\right]\right\}} \,. \tag{3.25}$$

The numerator in Eq. (3.25) gives the difference between the number of occurrence of U and the number of occurrence of D given C_i for a particular pattern h_i with sensitivity d when compared and matched with a time series across a rolling window with size w. Apart from that, we also have to take into account of the total number of occurrence of the pattern, which is included in the denominator in Eq. (3.25). A larger value of d would lead a larger number of occurrence of the pattern. The combination of h_i and d that gives the largest value of e_i would be chosen as Π_0; whereas the combination of h_i and d that gives the smallest value of e_i would be chosen as Π_1, or equivalently

$$(\Pi_0, d_0) = \arg\max_{(h_i,d)} e_i(d); \quad (\Pi_1, d_1) = \arg\min_{(h_i,d)} e_i(d) \,. \tag{3.26}$$

For (Π_0, d_0), we denote the ratio to be $E_0 = e_i(d_0)$, and for (Π_1, d_1), we denote the ratio to be $E_1 = e_i(d_1)$ with the requirement that for $E_0 > 0$ and $E_1 < 0$.

3.4 Trade Execution

After obtaining the likelihood ratio by estimating the probability density functions using the training set, and pattern selection of Π_0 and Π_1 with particular values of sensitivity d_0 and d_1 for evaluating the threshold, we can perform actions associated to U or D predictions, namely α_0 and α_1, when the patterns are matched with a particular segment in the testing set. Suppose our portfolio comprised of $M(t)$ units of capital and $N(t)$ shares of a particular stock, the objective is to increase the portfolio value (PV) with a finite number of actions. We also define four other variables $M_a(t)$, $M_b(t)$, $N_a(t)$, $N_b(t)$ that will be updated across time. Initially, the values are set to be $M_a = M_b = N_a = N_b = 0$ for all t . In our scheme, different investors may exhibit different attitude, which is characterized by the greed factor g. For investors adopting a more aggressive trading strategy, the value of g would be larger; whereas for investors who prefers less risky strategies would be associated with a smaller greed factor. In general, g is a non-negative real number no larger than unity: $0 \leq g \leq 1$.

In particular, if $\eta > \Lambda$, and C_0 is satisfied at $t = t'$, action α_0 is executed and the portfolio variables would be updated as follows:

$$M_a(t'+1) = 1 - gM(t'); \quad N_a(t'+1) = N(t') + \frac{gM(t')}{s(t')}\,, \tag{3.27}$$

and

$$M_b(t'+\Delta t) = (N_a(t'+1) - N(t'))s(t'+\Delta T); \quad N_b(t'+\Delta t) = -(N_a(t'+1) - N(t'))\,. \tag{3.28}$$

This corresponds to buying more shares when the investor holds bullish speculation and predicts that the stock price would increase. Therefore, the investor would buy more shares, which is reflected in the updating scheme of M_a and N_a. At $t = t' + \Delta t$, the investor would sell back the extra shares purchased at t', and this corresponds to the updating scheme of M_b and N_b.

On the other hand, if $\eta < \Lambda$, and C_1 is satisfied at $t = t'$, action α_1 is executed and the portfolio variables would be updated in the following way:

$$M_a(t'+1) = M(t') + gN(t')S(t'); \quad N_a(t'+1) = (1-g)N(t')\,, \tag{3.29}$$

and

$$M_b(t'+\Delta t) = -(M_a(t'+1) - M(t')); \quad N_b(t'+\Delta t) = \frac{M_a(t'+1) - M(t')}{s(t'+\Delta T)}\,. \tag{3.30}$$

This means when the investor speculates that the market is bearish, he or she will sell a portion of his or her shares at t', as denoted in the change of M_a and M_b. At $t = t' + \Delta t$, the investor would buy more shares with the extra units of capital obtained during t', and this corresponds to the updating scheme of M_b and N_b.

For other cases, the investor would not take any actions or implemented any manoeuvres to change the portfolio composition as he or she has no particular views on the near prospects of the stock market. This corresponds to the following updating rules:

$$M_a(t'+1) = M(t'); \quad N_a(t'+1) = N(t')\,, \tag{3.31}$$

and

$$M_b(t'+\Delta t) = 0; \quad N_b(t'+\Delta t) = 0\,, \tag{3.32}$$

After updating $M_a(t')$, $N_a(t')$, $M_b(t'+\Delta t)$ and $N_b(t'+\Delta t)$ according to the one of the above three scenarios, the portfolio is then updated by

$$M(t'+1) = M_a(t'+1) + M_b(t'+1); \quad N(t'+1) = N_a(t'+1) + N_b(t'+1)\,. \tag{3.33}$$

3.5 Implementation

Hang Seng Index (.HSI) is one of the major indices for the Hong Kong stock market composed of 50 constituents that are selected from the four sectors, namely Finance, Utilities, Properties and Commerce & Industry. We apply our method with Baysian signal detection and time warping to the closing price of Hang Seng Index since it is representative to the performance of the major companies in Hong Kong and is one of the commonly used indicators. In this experiment, we consider the time series of adjusted closing index price of Hang Seng Index from the period of 1987-11-16 to 2008-01-15. We treat the first 4000 closing prices, which roughly consists of trading data for around 16 years, to be the training set and the last 1000 closing prices to be the testing set. We have set the window size w to be 25 days, which is around 1-month time. Furthermore, the waiting time Δt is set to be 1 as it is one of the well-known stylized fact that the autocorrelation of closing price for stocks have short memory. For $\Delta t = 1$, the parameter for determining the digitization of rate of return into U and D is set to be $x_d = 0.003$. For larger Δt, x_d would also be larger.

We first select which chart pattern to use as Π_0 and Π_1 by obtaining the $e_i(d)$ from the training set, and plot out $e_i(d)$ for $0.02 \leq d \leq 0.04$ with 0.001-step increment in Fig. 3.7. Our criteria for the chart patterns is to select the patterns with maximal value of $|e_i(d)|$ while keeping the value of sensitivity d to be small. From Fig. 3.7, we would choose

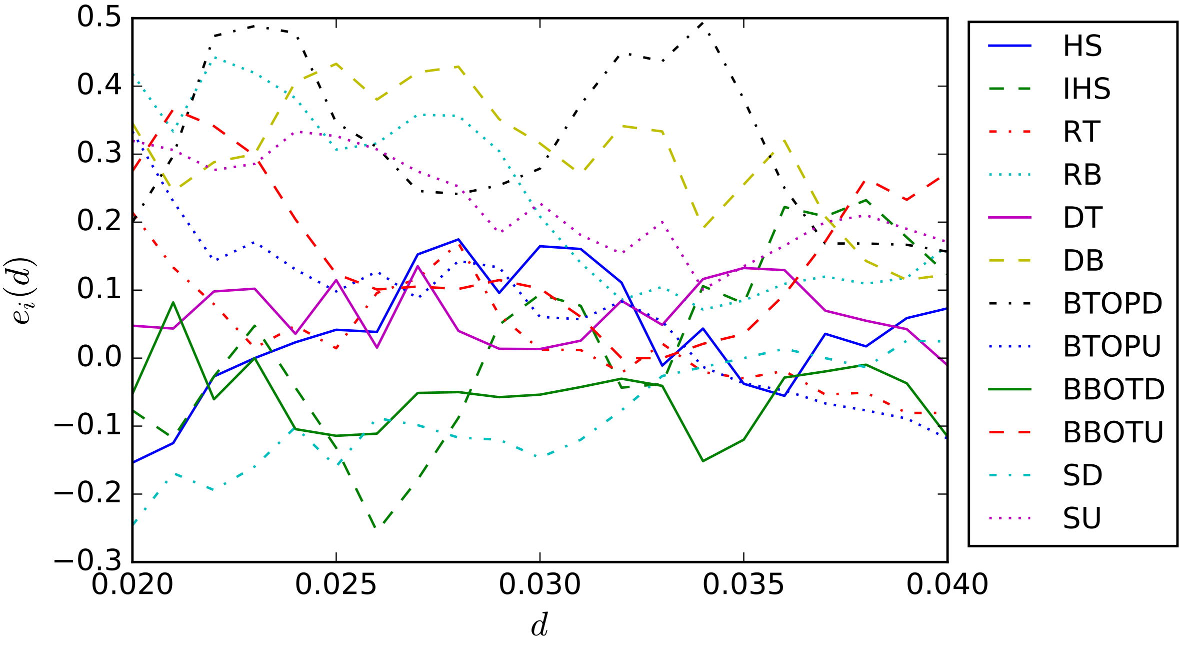


Figure 3.7: Graph of difference ratio $e_i(d)$ against sensitivity d for different chart patterns for the testing set of of adjusted closing price of Hang Seng Index

(BTOPD, 0.023) as (Π_0, d_0) , which has a difference ratio of $E_0 = 0.488$, and choose (SD, 0.02) as (Π_1, d_1) , which has a difference ratio of $E_1 = -0.246$. We are interested to first using chart patterns only when making investment decisions without invoking the use of Bayes detector, meaning that we only use the closing price for the past 25 days to predict whether there will be an event of U or D while information regarding rate of return is not used. This means we manage the portfolio based on C_0 and C_1 only for the price patterns without using Bayes detector on the rate of return. Because no transactions would involve trading of all assets due to Eq.(3.27), (3.29) we first set $g = 0.9$ for clearer illustration . The value for such portfolio is plotted in Fig. 3.8. It involves active management of the portfolio with 10 instances of α_0 and 13 instances of α_1, and it is shown that the portfolio with active management is worse than the portfolio without active management. As a comparison, we then use the chart patterns in combination with the rate of return using Bayes detector with approximation of pdf using sigmoid function, which has 7 instances of

α_0 and 11 executions of α_1, and plotted the fractional difference of corresponding portfolio values compared to index price.

Initially, the investor is endowed with $M(0) = 1 \times 10^9$ units of capital and $N(0) = 1 \times 10^5$ units of shares. Notice that PV in Fig. 3.8 refers to the the portfolio value that is scaled by the inverse of the initial value of the portfolio. Therefore, $PV(0) = 1$ for all portfolios. The quantities V_m and V_n are the values of the portfolio involving active management using chart patterns only evaluated in terms of units of capital and numbers of shares respectively, defined as

$$V_m(t) = M(t) + N(t)S(t) \quad \text{and} \quad V_n(t) = N(t) + \frac{M(t)}{S(t)}\,. \tag{3.34}$$

Similarly, the quantities W_m and W_n are the values of the portfolio without active management using chart patterns only evaluated in terms of units of capital and numbers of shares respectively. This means W_m and W_n are the values of portfolio with the buy-and-hold strategy. From Fig. 3.8, $V_m(t)$ is smaller than $W_m(t)$ for large t, while $V_n(t)$ is smaller than W_n, meaning the buy-and-hold strategy is better than using chart patterns alone, even though E_0 is positive and E_1 is negative. There are two plausible explanations. First, the sample size in the testing set is small, consisting of only 23 transactions. Second, we have not taken into account of the expected cost of actions corresponding to U and D.

On the other hand, PV_f in Fig. 3.9 refers to the fractional difference of value of portfolio with active management involving chart pattern and Bayes detector with rate of return V_m to the value of portfolio without active management.

$$PV_f = \frac{V_m - W_m}{W_m}\,. \tag{3.35}$$

The portfolio value is greater for portfolios with larger values of greed factor g. This is because Bayes detector has taken into account of the cost of actions and thus has excluded transactions that would otherwise lead to large decrease of portfolio values,. The total number of transactions now has dropped from 23 to 18. Notice however that if we only focus on a certain interval that corresponds to the drop of PV_f, larger values of g would obviously lead to larger decrease of the portfolio value compared the the ones with smaller g. A better way for actively managing the portfolio would be to use smaller values of g during those periods and to use larger values of g for instances that corresponds to correct predictions. We would develop such ideas in more details in the next chapter with

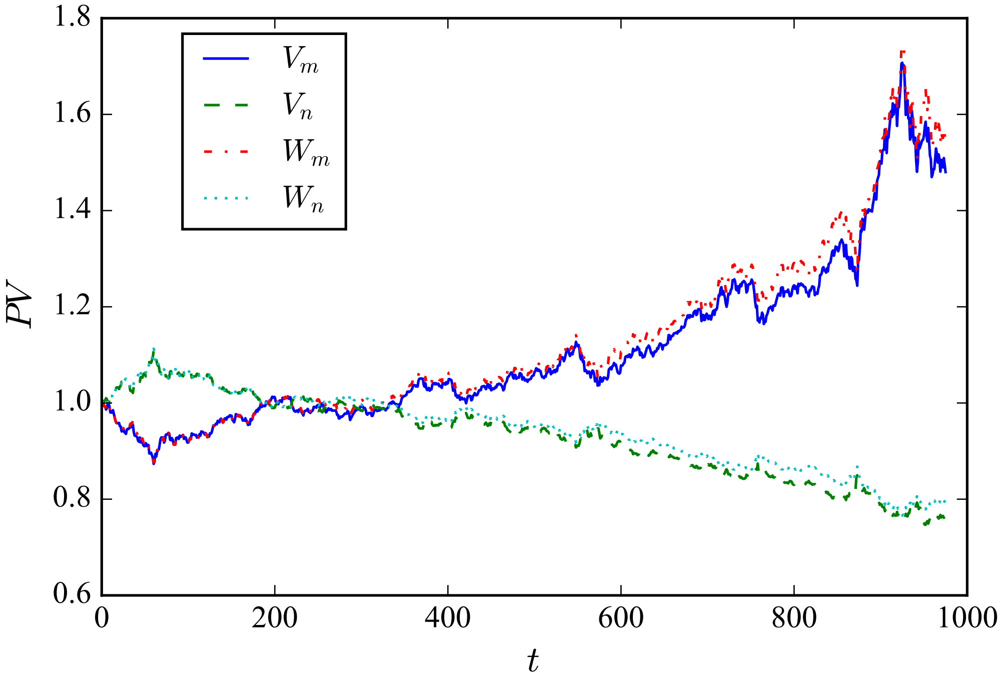


Figure 3.8: Value of portfolio PV across time t with recognition of chart patterns only for $g = 0.9$ with 10 instances of α_0 and 13 instances of α_1. V denotes value of portfolio with active management and W denotes value of protfolio without active management for HSI.

strategies consisting of a mixture of different values of greed factor, such that the greed factor is a function of time $g = g(t)$.

Apart from Hang Seng Index, we consider the closing price of the Financial Times Stock Exchange 100 index (.FTSE), which is composed of 100 UK companies on London Stock Exchange, obtained during the same period as HSI. For (Π_0, d_0), we simply use (RB,0.023); whereas for (Π_1, d_1), we use (SU,0.029). Fig. 3.7 shows the difference ratio of different patterns for the closing price of FTSE with $\Delta t = 1$. Idealistically, SD should be used since it attains the maximum of e at $d = 0.021$. However, because of the average cost estimated by SD is much larger than that by SU, we would use RB instead, which attains the second largest value of e among all patterns. The performances of portfolios with different values of g for FTSE is shown in Fig. 3.11, showing that the portfolio value increases with greed factor, which is similar to the portfolio involving HSI. The difference, however, is that there are more predictions of U than that of D because of the estimation of threshold and likelihood. Altogether, there are 23 executions of α_1 and 3 executions of α_0 for FTSE.

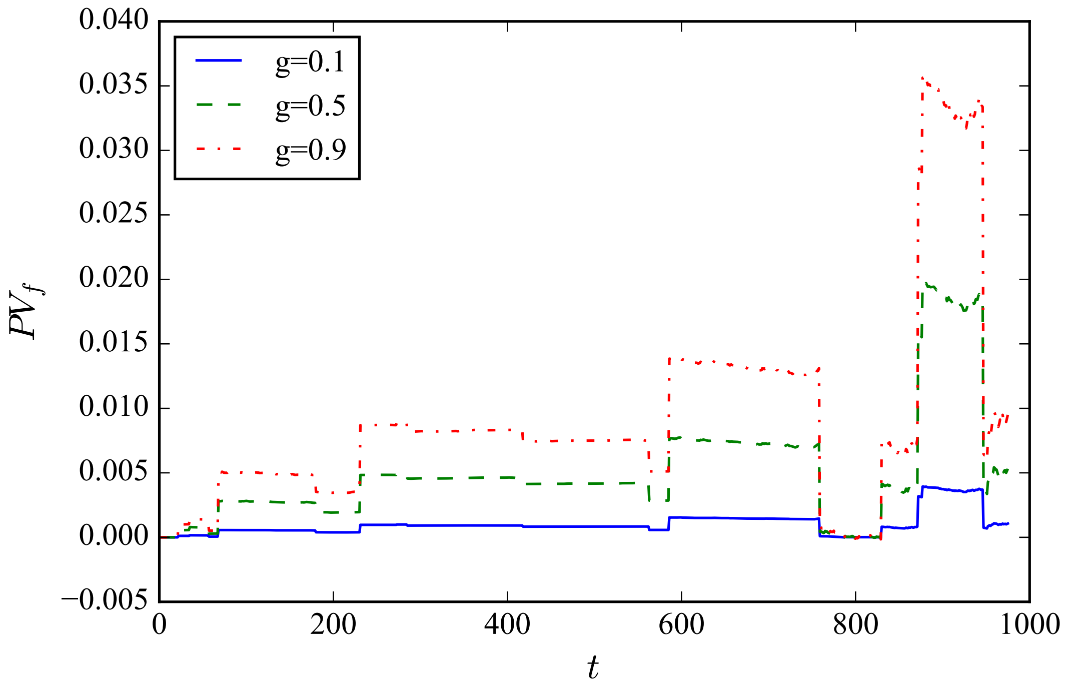


Figure 3.9: Fractional difference of portfolio value PV_f across time t with recognition of chart patterns and Bayes detector for rate of return with different values of g with 7 instances of α_0 and 11 instances of α_1 for HSI.

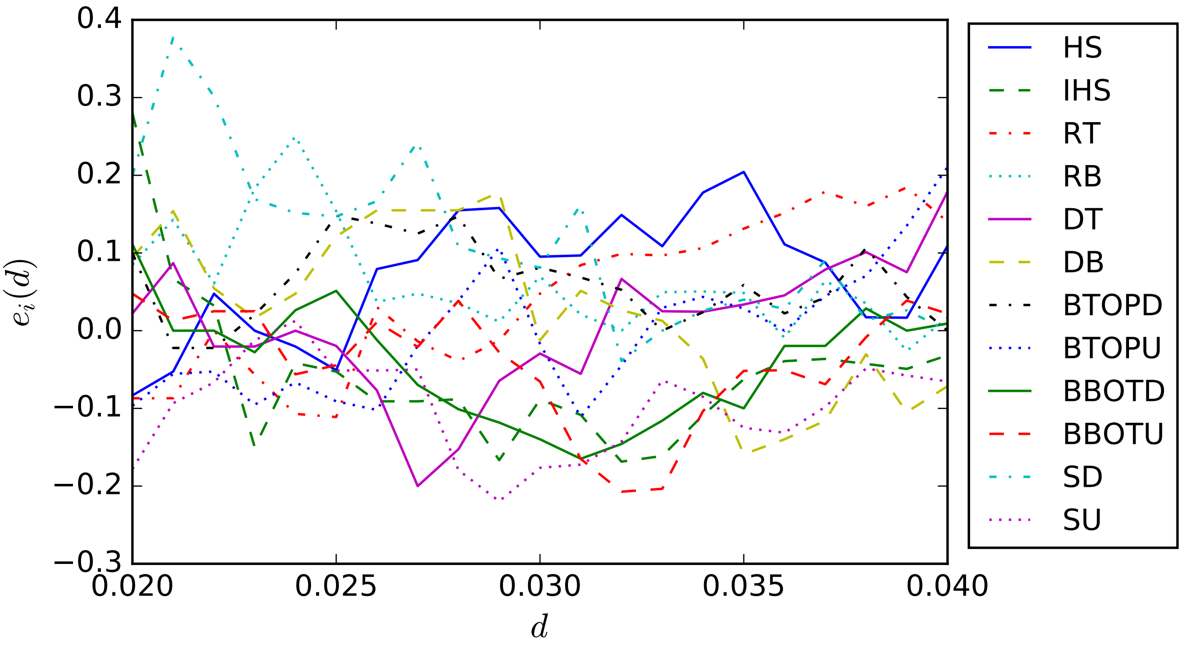


Figure 3.10: Graph of difference ratio $e_i(d)$ against sensitivity d for different chart patterns for the testing set of of adjusted closing price of FTSE

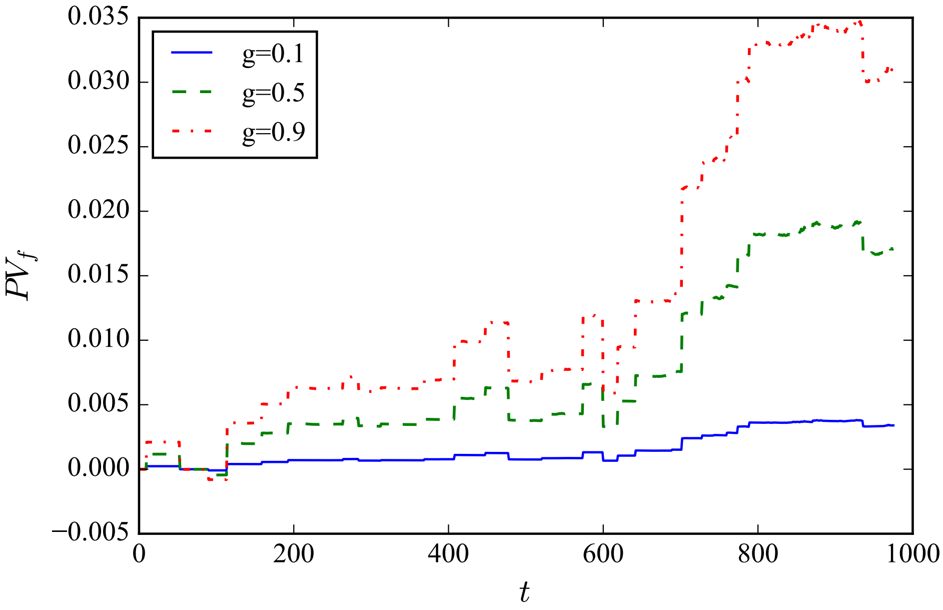


Figure 3.11: Fractional difference of portfolio value PV_f across time t with recognition of chart patterns and Bayes detector for rate of return with different values of g with 23 instances of α_0 and 3 instances of α_1 for FTSE.

3.6 k-means clustering

Previous sections have assumed the patterns of stock price may take a certain form that are commonly, qualitatively and subjectively used by technical analysts, and we try to use a more objective manner to incorporate such ideas into trading quantitatively. In this section, we do not assume such bias. Instead of imposing assumed patterns and comparing the similarities of these patterns with segments of the closing prices of stocks, we would like to discover patterns from the time series. Therefore, we turn our focus to unsupervised learning.

The simplest approach to unsupervised learning is k-means algorithm, which is described below [41, 42]. Suppose initially K initial positions of reference vectors $\mathbf{r}_i$are assigned randomly. Each data points in the set $\{\mathbf{x}^{(\ell)}\}_{\ell=0}^{N-1}$is classified or clustered to one of the K vectors, or cluster i, that has the smallest distance between $\mathbf{r}_i$ and $\mathbf{x}^{(\ell)}$.

$$i = \arg\min_j ||\mathbf{x}^{(\ell)} - \mathbf{r}_j|| \, , \tag{3.36}$$

and each data point is assigned with a label

$$b_j = \begin{cases} 1 & \text{if } j = i \\ \\ 0 & \text{else} \end{cases} . \tag{3.37}$$

The position of the reference points will then be updated by taking the average of positions using the labels

$$\mathbf{r}_i = \frac{\sum\limits_{\ell=0}^{N-1} b_i^{(\ell)} \mathbf{x}^{(\ell)}}{\sum\limits_{\ell=0}^{N-1} b_i^{(\ell)}} . \tag{3.38}$$

The steps involving Eq.(3.36), (3.37) and 3.38 are then iterated until the position of the reference points $\mathbf{b}_i$ converges, where $i = 0, 2, \ldots, K - 1$. The solution would always converge in finite number of steps. Trivially, to achieve minimal distance denoted by Eq. (3.36), we may simply set $K = N$, meaning that each cluster correspond to 1 point. On the other hand, K that are too small, the distance would be large. The number of reference points K is an inductive bias which determines the number of clusters. For simplicity, we have set $K = 10$ for the remaining of this section. More specifically, we use a rolling window with same window size $w = 25$ as previous section and sample segments of the time series at every 10 days as the input of k-means clustering. Altogether we minded 10 patterns, namely $k0, k1, \ldots, k9$. After normalizing the patterns obtained with k-means clustering, the difference ratio $e_i(d)$ is obtained from the training set of adjusted price of Hang Seng Index and is plotted in Fig. 3.12. If there exists any values of d in $0.02 \leq d \leq 0.04$ such that

$$\left[\sum_{x^{(\ell)} \in R_i} (\alpha^{(\ell)})^2 \right] + \left\{ \sum_{x^{(\ell)} \in R_i} \left[(\alpha^{(\ell)} - 1)^2 - \gamma^{(\ell)} \right] \right\} = 0 \, , \tag{3.39}$$

we would neglect that particular pattern ki and would not plot the difference ratio $e_i(d)$ for ki in Fig. 3.12. This is because it implies that pattern ki does not give good prediction of either U or D for small values of d. From Fig. 3.12, we choose (k1,0.035) as $(\Pi_0, d_)$ with $E_0 = 0.266$ and (k4,0.028) as (Π_1, d_1) with $E_1 = -0.488$. The corresponding threshold is always smaller than the likelihood ratio, meaning the investor should always sell and never buy shares. We therefore change (Π_1, d_1) to be (k4,0.036) $E_1 = -2.323$, which is the second smallest value in the graph. The patterns of $k2$ and $k4$ are shown in Fig .3.13 and 3.14. It is interesting to notice that some prominent features of pattern $k2$ as shown

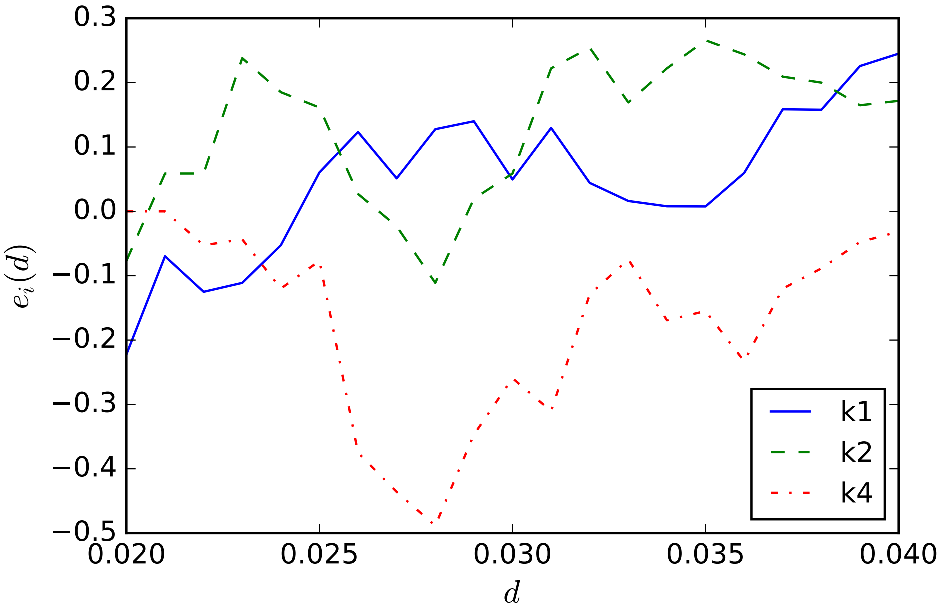


Figure 3.12: Graph of difference ratio $e_i(d)$ against sensitivity d for different patterns obtained with k-means algorithm for the testing set of adjusted closing price of Hang Seng Index

in Fig. 3.13 is quite similar with the features of the broadening top patterns (BTOP) as shown in Fig. 3.4 and 3.5. As mentioned, BTOP belongs to the category of trend continuation patterns, and [39] defines BTOP by a successive of alternate local extrema E_1, E_2, E_3, E_4 and E_5 with

$$BTOP = \begin{cases} E_1 \text{ is a maximum} \\ E_1 \leq E_3 \leq E_5 \\ E_2 \geq E_4 \end{cases}, \tag{3.40}$$

with ascending broadening-tops and descending broadening-tops being two of the special cases. Notice that we have made a slight deviation from the definition given by [39] by replacing the strict inequalities with non-strict inequalities.

There are altogether 22 actions corresponding to U and 19 actions corresponding to D. The fractional difference of portfolio value PV_f defined in Eq. (3.35) is plotted in Fig. 3.15. For large t, the portfolios with large greed factor g outperforms the ones with small g. Nonetheless, if we only focus on the period before $t = 400$, portfolios with small values of g performs better. It is desirable to use s small value of g for less accurate predictions and use a large g for more accurate predictions. Such strategies will be discussed in the

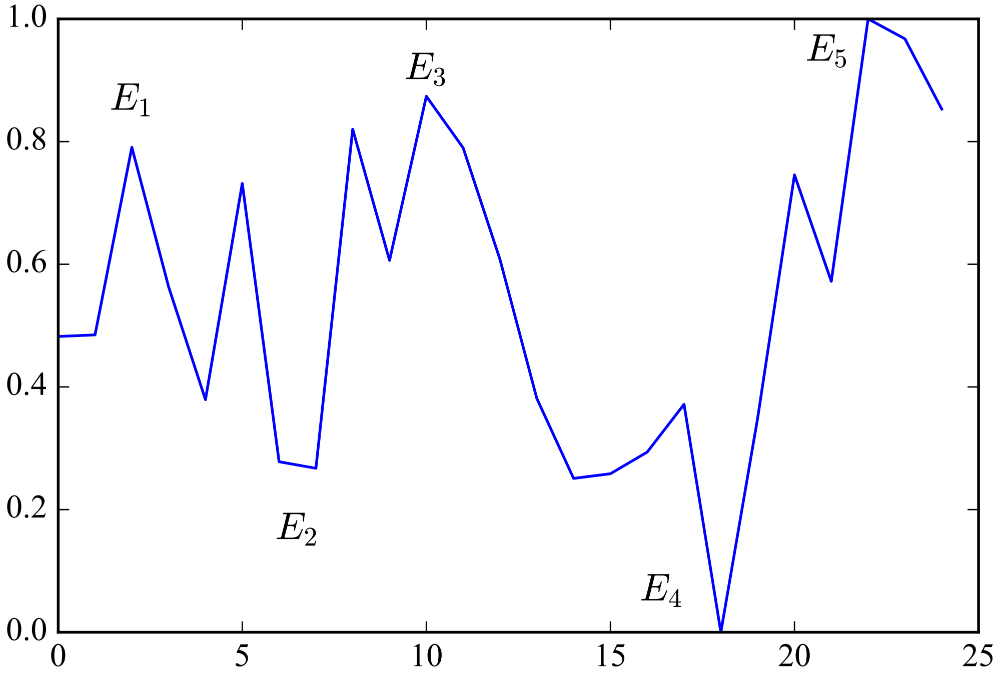


Figure 3.13: Pattern $k2$ obtained with k-means algorithm with $w = 25$ and $K = 10$ after normalization. The pattern is used as Π_0 for trading that corresponds to bullish speculations.

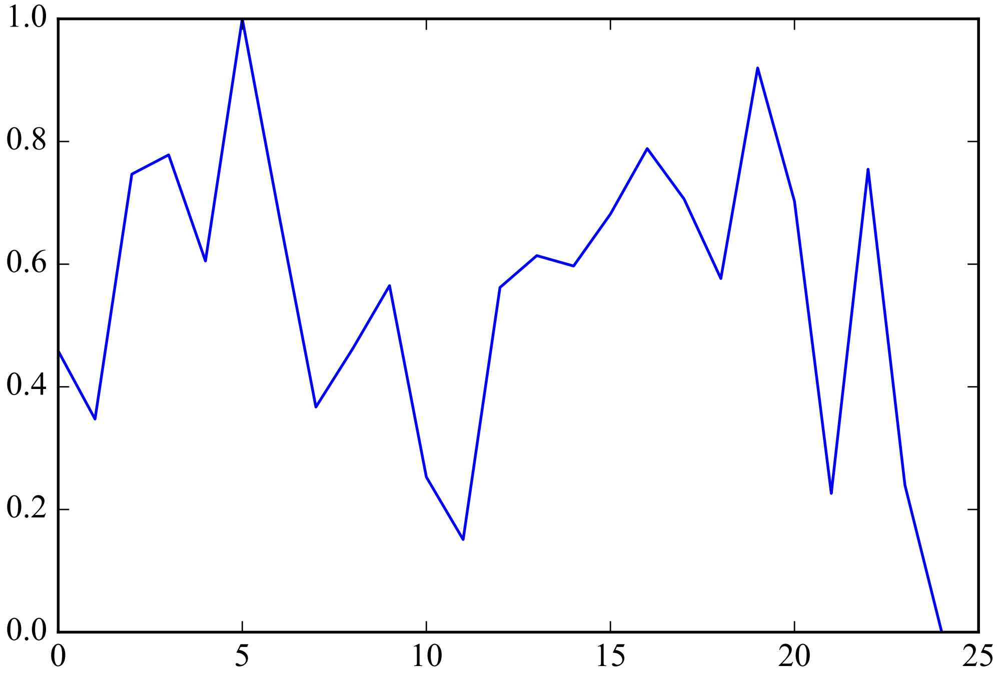


Figure 3.14: Pattern $k4$ obtained with k-means algorithm with $w = 25$ and $K = 10$ after normalization. The pattern is used as Π_1 for trading that corresponds to bearish speculations

next chapter.

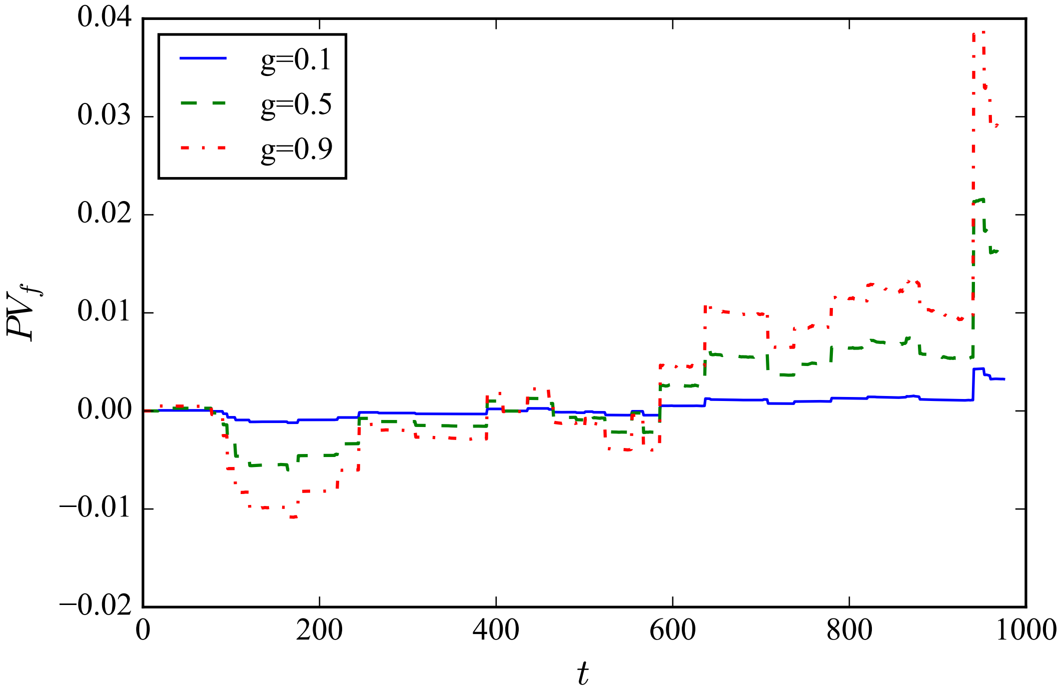


Figure 3.15: Fractional difference of portfolio value PV_f for HSI across time t with recognition of chart patterns and Bayes detector for rate of return with different values of g with 22 instances of α_0 and 19 instances of α_1.

We then apply the algorithm to daily closing price of FTSE. Same as before, we plotted out the difference ratio of the patterns in Fig. 3.16. Here, we use pattern $k3$ to be Π_0 since it corresponds to the highest e that is smaller than one. Similarly, we choose pattern $k8$ to be Π_1. Nonetheless, at $d_0 = 0.022$ and $d_1 = 0.022$, there is a large difference between the expected costs corresponding to Π_0 and Π_1, such that the threshold is always smaller than the likelihood ratio. Therefore, we use $d_1 = 0.023$ instead. The portfolio value at different values of greed factor g is then plotted in Fig. 3.17.

3.7 Volume Patterns

So far we have investigated patterns in index price. By using information in historic price, we attempt to make rational decisions of investments. Nevertheless, historic price is not the only available information to the public and investors may be interested to make use of a combination of information sources in order to make a better-informed decision. For

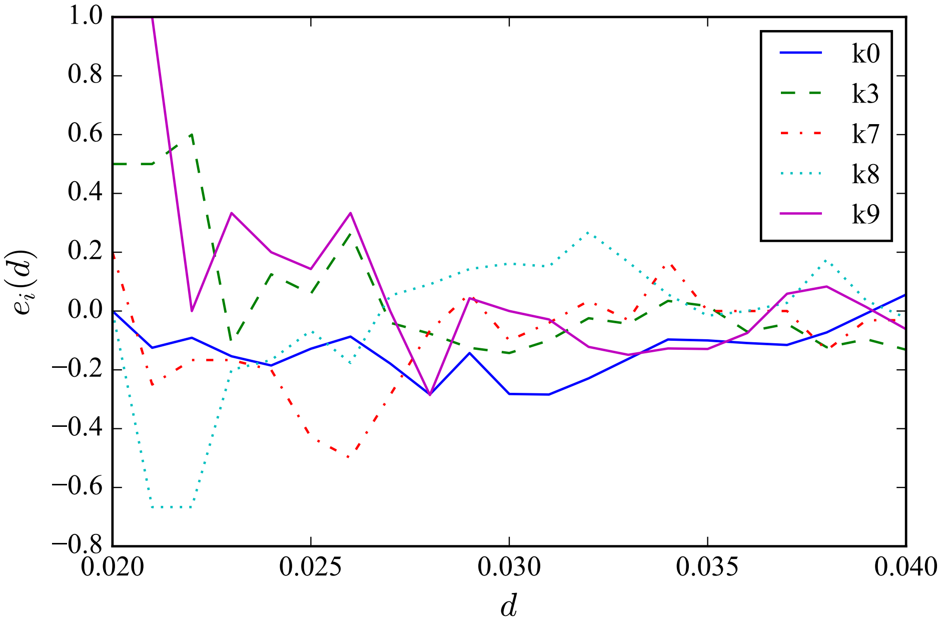


Figure 3.16: Graph of difference ratio $e_i(d)$ against sensitivity d for different patterns obtained with k-means algorithm for the testing set of adjusted closing price of FTSE

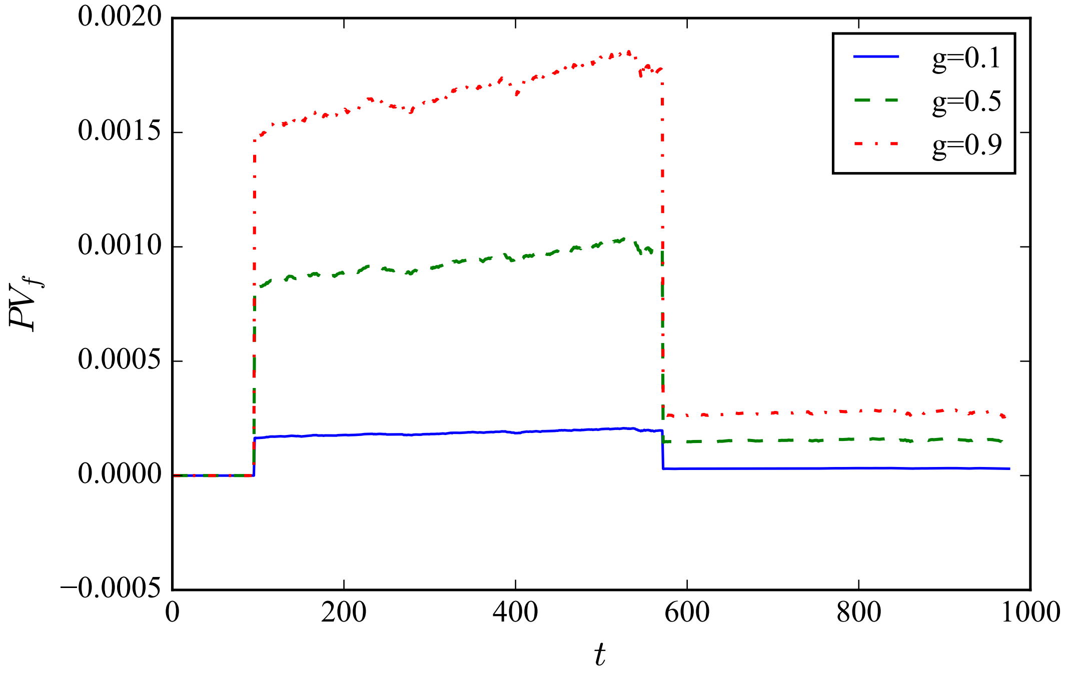


Figure 3.17: Fractional difference of portfolio value PV_f for FTSE across time t with recognition of chart patterns and Bayes detector for rate of return with different values of g with 2 instances of α_1.

example, we can incorporate and consider the information of both the index price and volume before determining whether if we should buy or sell a particular amount of shares. In Sect. 3.2, we have introduced a total of 12 commonly used technical patterns that can be applied to one time series. For two time series, namely the adjusted daily closing price and daily volume, there is a combination of $12 \times 12 = 144$ possible cases, and each case is assigned with a case number as shown in Table 3.1. We denote the patterns for the index price to be Π_i^p and the patterns for the volume to be Π_j^v, where $i,\ j = 1, 2, \ldots, 12$. Consider the time series of index price $s(t)$ and index volume $v(t)$. At $t = t'$, the segment $v_{t'} : \{v(t'-w+1) : v(t')\}$ is observed to decide the investing in $s(t)$. Condition is satisfied at $t = t'$ if the following relation is true:

$$C_k : D_p(\Pi_k^p, s_{t'}) \leq d \wedge D_v(\Pi_k^v, v_{t'}) \leq d\,. \tag{3.41}$$

Due to the increased number of cases, the probability of each case $P(\Pi_i^p, \Pi_j^v)$ in the training set would be reduced compared to the $P(\Pi_i)$ used in previous sections, which can be obtained by marginalization. Moreover, due to the availability of data, we will only concern ourselves with the data from 2001-08-13 to 2015-12-31, with 3600 trading days. The first 2600 days are set to be training set, while the remaining 1000 data points are the testing set. Because of the ratio of size of the training set to the size of testing set, we do not expect the prediction rules extracted from the training set to be very accurate. Moreover, we can set the sensitivity to be a rather large fixed number, $d = 0.04$, so that the number of occurrence for each case would not be too small. Similar to previous section, the window size is $w = 25$ and number of occurrence of U and the number of occurrence of D for all cases are plotted. Because the value of d is fixed, we used the follow quantity to instead of the e ratio when selecting patterns:

$$N_I - N_D = \left[\sum_{x^{(\ell)} \in R_i} (\alpha^{(\ell)})^2 \right] - \left\{ \sum_{x^{(\ell)} \in R_i} \left[(\alpha^{(\ell)} - 1)^2 - \gamma^{(\ell)} \right] \right\} \tag{3.42}$$

From Fig. 3.18, we determine to use case 1 and case 36 as the prediction rules, which has $N_U - N_D$ to be -13 and 23 respectively. More specifically, we have $(\Pi_0^p, \Pi_0^v) = (HS, IHS)$ and $(\Pi_1^p, \Pi_1^v) = (RB, HS)$ with $d = 0.04$ to be the prediction rules. Setting $g = 0.9$, we compare the value of the portfolio using patterns of both price and volume $(\Pi_0^p, \Pi_0^v, \Pi_1^p, \Pi_1^v, d)$ to that of the portfolios using only price pattern (Π_0^p, Π_1^p, d) or volume

Table 3.1: Case numbers for all possible combinations of chart patterns in prices with chart patterns in volume

price\vol	HS	IHS	RT	RB	DT	DB	BTOPD	BTOPU	BBOTD	BBOTU	SD	SU
HS	0	1	2	3	4	5	6	7	8	9	10	11
IHS	12	13	14	15	16	17	18	19	20	21	22	23
RT	24	25	26	27	28	29	30	31	32	33	34	35
RB	36	37	38	39	40	41	42	43	44	45	46	47
DT	48	49	50	51	52	53	54	55	56	57	58	59
DB	60	61	62	63	64	65	66	67	68	69	70	71
BTOPD	72	73	74	75	76	77	78	79	80	81	82	83
BTOPU	84	85	86	87	88	89	90	91	92	93	94	95
BBOTD	96	97	98	99	100	101	102	103	104	105	106	107
BBOTU	108	109	110	111	112	113	114	115	116	117	118	119
SD	120	121	122	123	124	125	126	127	128	129	130	131
SU	132	133	134	135	136	137	138	139	140	141	142	143

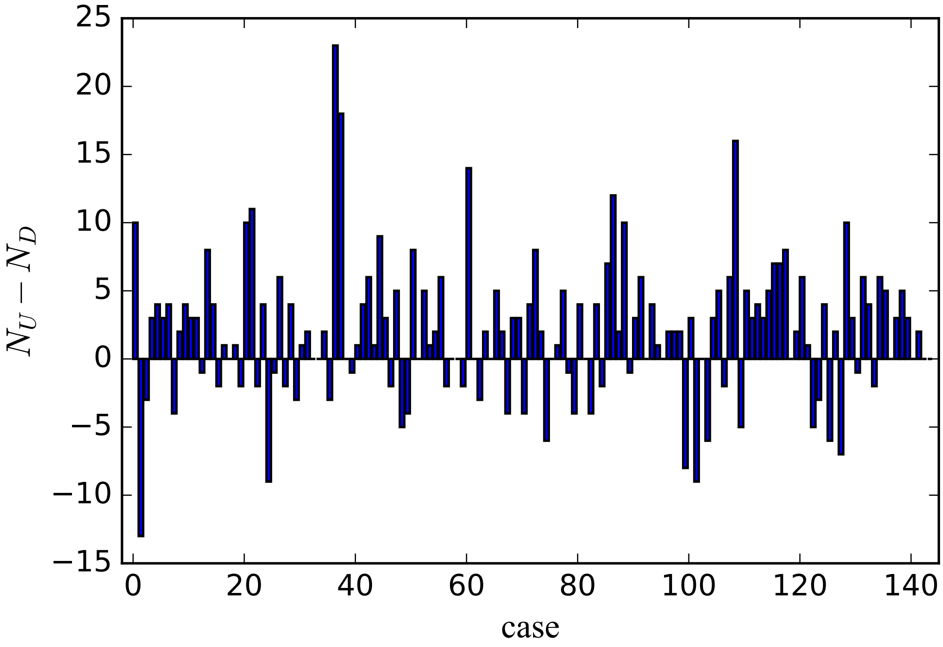


Figure 3.18: The difference between number of U to number of D for all possible cases for HSI.

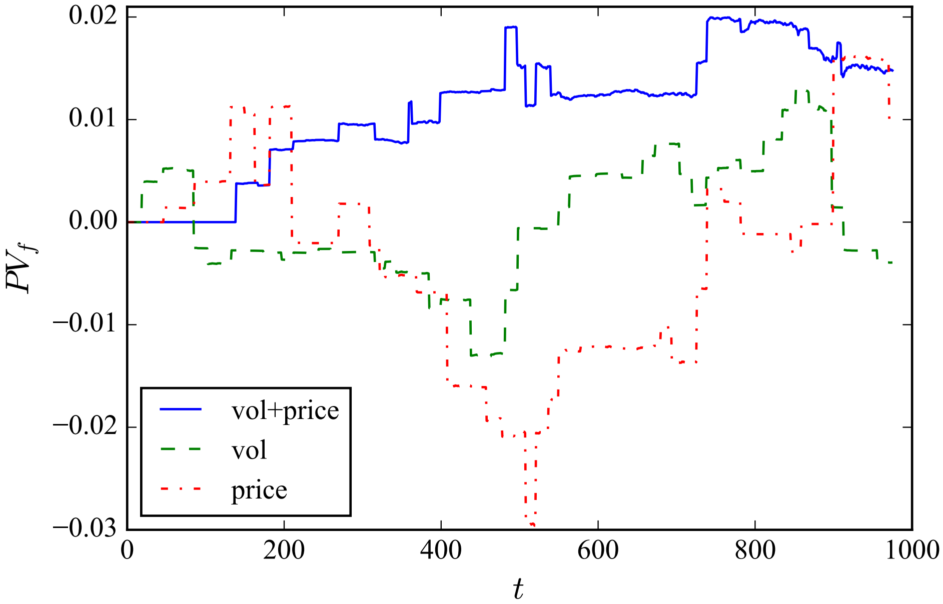


Figure 3.19: Fractional difference of portfolio value PV_f for HSI across time t with Bayes detector and recognition of chart patterns for historic price (red), historic volume (green) and both historic price and volume (blue).

pattern (Π_0^v, Π_1^v, d) and plot the fractional difference with respect to index Fig 3.19.

For validation, we again use the closing price of FTSE. Due to the availability of historic data of volume, we are only concerned with the period from 04-01-1999 to 31-12-2015, consisting of 4295 trading days. Same as before, we would use the last 1000 days as the testing set, while the first 3295 days are used as training set. From the training set, we obtain the difference between number of occurrence of U and number of occurrence of D. We use $(\Pi_0^p, \Pi_0^v) = (BBOTU, IHS)$ and $(\Pi_1^p, \Pi_1^v) = (DT, DB)$ with $d = 0.04$, corresponding to the value of $N_U - N_D$ to be 10 and -19 respectively. The histogram is plotted in Fig. 3.20, and the portfolio performances are shown in Fig. 3.21. Initially, the portfolio using only historic price for prediction outperforms the portfolio using both information of price and volume, the situation is reversed for $t > 800$. Notice that the latter case has much less transactions compared to the prior case.

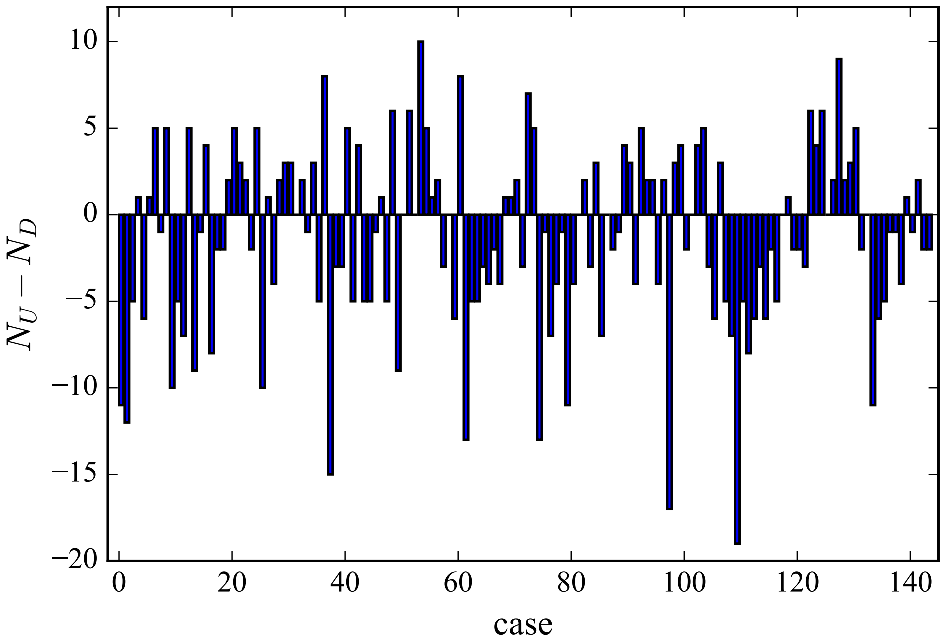


Figure 3.20: The difference between number of U to number of D for all possible cases for FTSE.

3.8 Summary

In this section, we first introduce Signal Detection Theory and discuss using SDT on portfolio management. On the other hand, we show some of the most commonly used chart patterns in technical analysis. Using SDT on rate of return for stocks and DTW for patterns, we conduct active management of our portfolio, which consists only of one

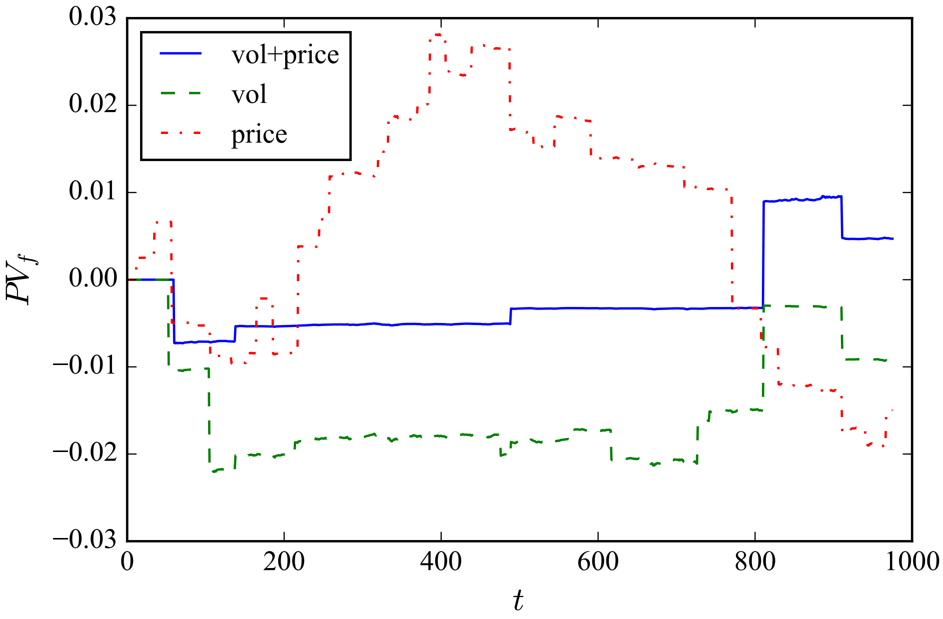


Figure 3.21: Fractional difference of portfolio value PV_f for FTSE across time t with Bayes detector and recognition of chart patterns for historic price (red), historic volume (green) and both historic price and volume (blue).

risky asset with cash, and it is shown that the combination of SDT and DTW performs better than using DTW only. In terms of consistency of the results, we notice that portfolios with larger greed factor performs better for both data sets of HSI and FTSE. Moreover, we have also demonstrate the usefulness of our method using a different data set of a different period in the final section involving volume patterns. However, when considering a different time scale or a different size for the testing set, our methodology would not be as useful if the size is too small since the number of transactions would be too small and such events may not be statistically significant. Therefore, we adopt in our work a window size of 25 days for the rolling window and a testing set with 1000 trading days. This shows the effectiveness of our trading scheme. We then use k-means algorithm instead of predefined patterns for rule-based prediction, which also yields comparable percentage of returns. In the final section, patterns in time series of historic volume is considered in addition to the daily closing price. We demonstrate that using both historic patterns and price patterns in trading is better than only using historic patterns or price patterns.

Chapter 4

Parrondo Effect

In this chapter, the setting of the original Parrondo's game would be briefly introduced. Using some simple non-linear time-series as a toy model, we then introduce two games characterized by two investment strategies which demonstrates Parrondo's effect by combining two losing investment strategies into a winning strategy. The similar methodology is then applied on real stock data with chart patterns. Inspired by Parrondo's effect in the toy model, fuzzy signal detection theory is introduced and applied on portfolio management in the final section .

4.1 Parrondo Games

Parrondo games refers to the combination of two losing games into a winning game. Initially, the player have zero units of capital. The player would receive one unit of capital if he or she wins that round of the game. If the player loses, a unit of capital will be taken from him or her for that round of the game. Let us denote the amount of capital to be $C(t)$, which would evolve as the game progresses.

The original Parrondo game consists of two games, namely game A and game B. Game A involves flipping a biased coin, which gives a winning probability p_A, where p_A is smaller than half. Game B is a state-dependent game involving two biased coins, with varying winning probability p_B. The state of the player is defined by the amount of capital he or she possesses. If C is a multiple of M, which is an integer, the winning probability will be $p_B = p_b$; if C is not a multiple of M, $p_B = p_g$ will be the winning probability instead. The original setting of the parameters are $p_A = 0.5 - \epsilon$, $p_g = 0.75 - \epsilon$, $p_b = 0.1 - \epsilon$ and

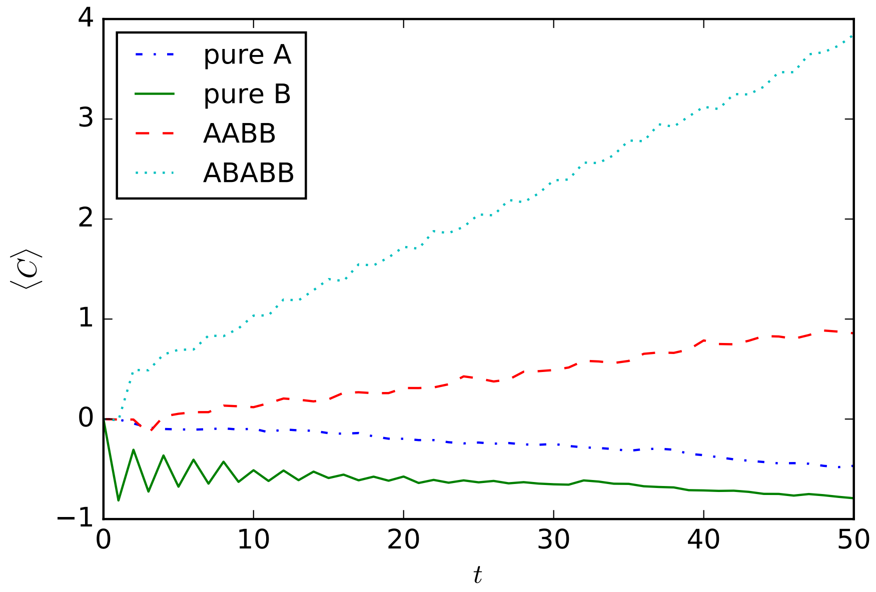


Figure 4.1: Evolution of average capital $\langle C \rangle$ against time t over 5000 realizations with $\epsilon = 0.003$ for different game sequences: pure game A sequence (blue), pure game B sequence(green), AABB (red) and ABABB (cyan).

$M = 3$, where ϵ is some small real number. If $\epsilon = 0$, then both game A and game B are fair games.

$$p_A = 0.5 - \epsilon; \quad p_B = \begin{cases} 0.10 - \epsilon & \text{if } C\%3 = 0 \\ 0.75 - \epsilon & \text{else} \end{cases} \tag{4.1}$$

There are different ways of combining two games, such as random mixing and periodic switching. Random mixing refers to randomly choosing game A and game B with a certain probability. Periodic switching refers to selecting which game to play based on a predetermined periodic game sequence. In this work, I will focus on the latter case. In Fig. 4.1, I have shown the capital gained by the player using pure game sequences and periodic sequences as a demonstration.

4.2 A Toy Model

Here, we formulate game A and game B in the context of investment, and use two simple examples to illustrate Parrondo effect.

4.2.1 Formulation

In the context of investment, suppose there are two investment strategies: strategy A and strategy B. Strategy A is a more risky strategy involving a large greed factor g_A, while strategy B is a more conservative investment strategy involving a smaller greed factor g_B such that $g_A > g_B$. For simplicity, we define k' to be the indice that denotes the games such that $k' = 0$ for game A and $k' = 1$ for game B while the greed factors are denoted as $g_{k'}$. Similar to the previous section, the daily rate of return, defined as the fractional change of stock prices $S(t)$ and $S(t-1)$, undergoes triplet quantization.

$$x_q = \begin{cases} U & \text{if } x \geq x_d \\ D & \text{if } x < -x_d \\ H & \text{else} \end{cases} \tag{4.2}$$

In addition, we assume the investor has m-step memory, meaning the information available to him or her at $t = t'$ are $\{x_q(t'), x_q(t'-1), ..., x_q(t'-m+1)\}$. The investor would update his or her portfolio composition with Eq. (3.27), (3.29) in previous section with $g_0 = 0.7$ and $g_1 = 0.3$. Then, the expected value of $C(t)$ and and $V(t)$are

$$\begin{aligned} \langle C(t+1) \rangle = & \left(\sum_{i=1}^{3^m} a_i P(U|C_i) \right) [(1-g_{k'})C(t)] \\ & + \left(\sum_{i=1}^{3^m} a_i P(D|C_i) \right) [C(t) + g_{k'} N(t) S(t)] + \left(\sum_{i=1}^{3^m} a_i P(H|C_i) \right) C(t) \,, \end{aligned} \tag{4.3}$$

$$\begin{aligned} \langle N(t+1) \rangle = & \left(\sum_{i=1}^{3^m} a_i P(U|C_i) \right) \left[N(t) + \frac{g_{k'} C(t)}{S(t)} \right] \\ & + \left(\sum_{i=1}^{3^m} a_i P(D|C_i) \right) [(1-g_{k'})N(t)] + \left(\sum_{i=1}^{3^m} a_i P(H|C_i) \right) N(t) \,, \end{aligned} \tag{4.4}$$

where a_i are indicator variables such that

$$a_i = \begin{cases} 1 & \text{if condition is } C_i \\ 0 & \text{if condition not } C_i \end{cases} \tag{4.5}$$

with

$$\sum_i a_i = 1 \,. \tag{4.6}$$

Now, we let $x_d = 0$, so that $P(H|C_i) = 0$ for all i and $a_j = 0$ for C_j that involves H. As a result, there is a reduction of cases from 3^m to 2^m cases with non-zero a_i. For 1-step memory, there are only 2 cases, namely U and D. Eq. (4.3), (4.4) are then reduced to

$$\begin{aligned}\langle C(t+1)\rangle = {} & [a_U P(U|U) + (1-a_U)P(U|D)[(1-g_{k'})C(t)] \\ & + [a_U P(D|U) + (1-a_U)P(D|D)][C(t) + g_{k'}N(t)S(t)]\,,\end{aligned} \tag{4.7}$$

$$\begin{aligned}\langle N(t+1)\rangle = {} & [a_U P(U|U) + (1-a_U)P(U|D)]\left[N(t) + \frac{g_{k'}C(t)}{S(t)}\right] \\ & + [a_U P(D|U) + (1-a_U)P(D|D)]\,[(1-g_{k'})N(t)]\ .\end{aligned} \tag{4.8}$$

At t, the portfolio value, which depends on the strategy used at that instance with the associated $g_{k'}$, can therefore be evaluated in terms of capital by

$$\begin{aligned}X_{g_{k'}} & \equiv \langle V(t+1)\rangle_{g_{k'}} \\ & = \left(\sum_{i=1}^{2^m} a_i P(D|C_i)\right)[C(t) + g_{k'}N(t)S(t) + (1-g_{k'})N(t)S(t+1)] \\ & + \left(\sum_{i=1}^{2^m} a_i P(U|C_i)\right)[(1-g_{k'})C(t) + [(N(t) + \frac{g_{k'}C(t)}{S(t)}]S(t+1))]\,.\end{aligned} \tag{4.9}$$

For $m = 1$, Eq. (4.9) is reduced to

$$\begin{aligned}X_{g_{k'}} = {} & (a_U P(D|U) + (1-a_U)P(D|D)) \\ & \times [C(t) + g_{k'}N(t)S(t) + (1-x)N(t)S(t+1)] \\ & + (a_U P(U|U) + (1-a_U)P(U|D)) \\ & \times [(1-x)C(t) + [(N(t) + \frac{g_{k'}C(t)}{S(t)}]S(t+1))]\,.\end{aligned} \tag{4.10}$$

Suppose strategy k' is adopted at a particular instance, the investor would evaluate the portfolio values. If $X_{g_{k'}} > X_{g_{(1-k')}}$, the investor would gain one extra token; whereas if $X_{g_{k'}} < X_{g_{(1-k')}}$, the investor would lose one extra token. The number of tokens the investor possesses is $M(t)$. Initially, the investor does not have any tokens, meaning $M(0) = 0$.

4.2.2 Sinusoidal Function

We first consider sinusoidal function with frequency f as the stock price for simplicity, and take $\Delta t = 1$ for sampling, which is similar to considering daily closing price for stocks. So the sampling frequency is 1Hz, while the function we consider is

$$s(n) = \cos(2\pi f n\Delta t) + 2\,. \tag{4.11}$$

We consider $\{s_n\}_{n=0}^{4002}$ to be the training set and $\{s_n\}_{n=5002}^{6002}$ to be the testing set. The conditional probabilities $P(U|C_i)$ and $P(D|C_i)$ are extracted from the training set, and the player plays the game to either win or lose a token per round according the the testing set. For 1-step memory, the difference of gain of token for both game A and game B is very small around frequency $f_a = 0.028655$, for which the sampling rate we adopt is well above the Nyquist rate. As shown from Fig.4.2 for $f = 0.03$, both are losing games, but the combination of the two games using periodic game sequence of AABB would lead to positive gain of tokens, showing the existence of Parrondo effect in this artificial example.

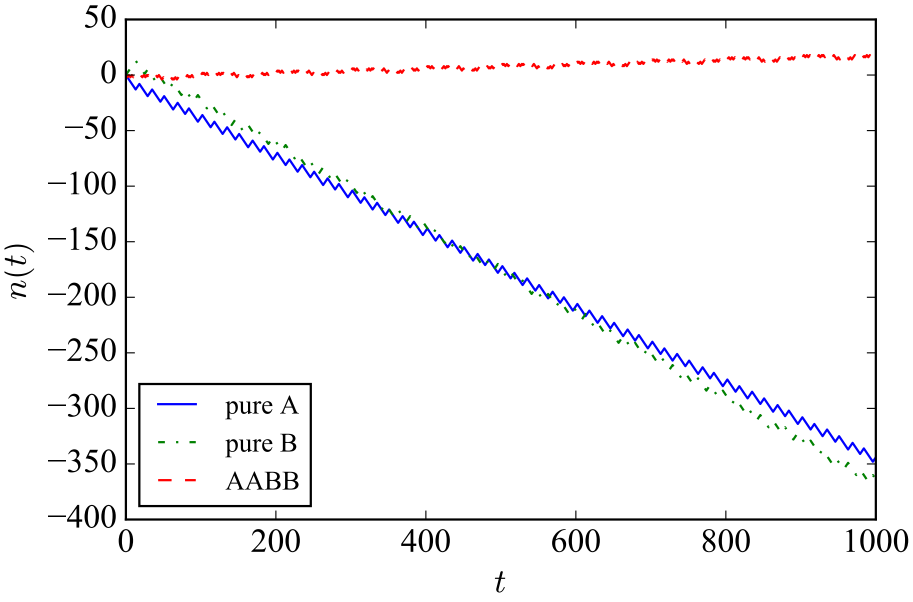


Figure 4.2: Graph of number of tokens n against time t demonstrating Parrondo effect in toy model of investment involving sinusoidal as input time series with $f = 0.03$. Both pure A sequence (blue) and pure B sequence (green) are losing strategies, while periodic sequence AABB (red) is a winning strategy with positive gain of capital.

With no memory, $P(U) = P(D) = 0.5$ due to symmetry, and the entropy for the daily return to be U or D is simply

$$H(s) = -P(U)\log P(U) - P(D)\log P(D) = 1\,. \tag{4.12}$$

For $f = 0.03$ with 1-step memory, $P(U|U) = P(D|D) = 0.94$ and $P(D|U) = P(U|D) = 0.06$, with the latter two conditional probabilities corresponding to the turning points of the sinusoidal. Given the condition to be U, the entropy is

$$H(s|U) = -P(U|U)\log P(U|U) - P(D|U)\log P(D|U) = 0.3274 \tag{4.13}$$

Due to symmetry, $H(s|U) = H(s|D)$. The conditional entropy is then

$$\begin{aligned} H(s|C) &= -\sum_i \sum_j P(s_i, C_j)\log P(s_i|C_j) \qquad (4.14)\\ &= -\sum_i P(s_i)\sum_j P(s_i|C_j)\log P(s_i|C_j)\,, \qquad (4.15)\end{aligned}$$

so $H(s|C) = 0.3274$. Then, with 1-step memory, the information gained can be quantified as $H(s) - H(s|C) = 0.673$. For frequencies smaller than f_a, the number of tokens n for pure B sequences is larger than that of pure A sequence. For frequencies larger than f_a, the number of tokens n for pure B sequences becomes smaller than that of pure A sequence. Such effect for AABB sequence disappears around $f_b = 0.06549$. For frequencies larger than f_b, the number of tokens for pure game A sequence is larger than that of AABB sequence. Notice that at $f = 0.5$, $P(D|U) = P(U|D) = 1$ and $P(U|U) = P(D|D) = 0$ as expected. Therefore, $H(s) - H(s|C) = 1$.

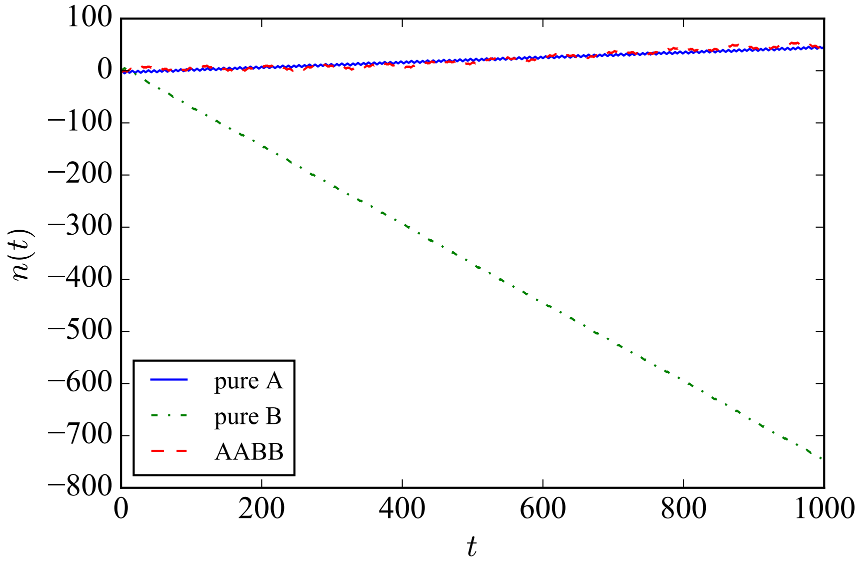


Figure 4.3: Graph of number of tokens n against time t demonstrating the disappearance of Parrondo effect in toy model of investment involving sinusoidal as input time series with $f = 0.06549$. While pure B sequence (green) is a losing strategy, Both pure A sequence (blue) and periodic sequence AABB (red) is a winning strategy with positive gain of capital.

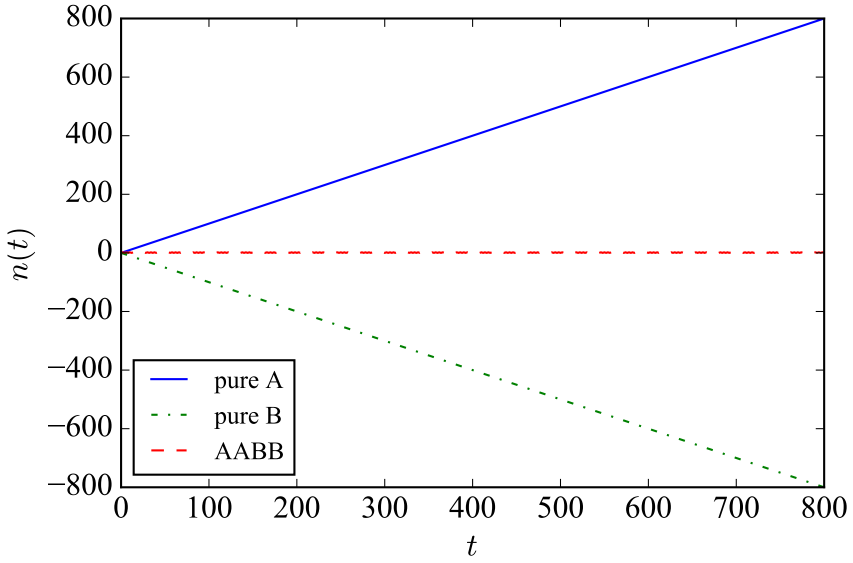


Figure 4.4: Graph of number of tokens n against time t in toy model of investment involving sinusoidal as input time series with $f = 0.5$. Pure A sequence (blue) is a winning strategy, pure B sequence (green) is a losing strategy, and periodic sequence AABB (red) is a fair strategy.

4.2.3 Lorenz System

Consider the Lorenz equations:

$$\frac{dx}{dt} = \sigma(y - x) \tag{4.16}$$

$$\frac{dy}{dt} = x(\rho - z) - y \tag{4.17}$$

$$\frac{dz}{dt} = xy - \beta z \tag{4.18}$$

The initial conditions (x_o, y_o, z_o) are $(2, 3, 4)$, while the parameters are set to be $\sigma = 10$ and $\beta = 8/3$, and the step size is set to be 0.01 for numerical integration from t=0 to t=200 . Therefore, there are altogether 20000 data points, and we define this data set to be $\{z^{(\ell)}\}_{\ell=0}^{20000-1}$ Under these parameters, Fig.4.5 shows the lorenz attractor projected onto xz-plane. Here, we treat $z(t)$ as the stock price, with ρ set to be 38, which corresponds to chaotic regime. We take $\{z^{(\ell)}\}_{\ell=4000}^{6002}$ to be training set and $\{z^{(\ell)}\}_{\ell=7000}^{8002}$ to be testing set. Using the formulation Sect.4.2.1, we demonstrate the existence of Parrondo effect as shown in Fig.4.6

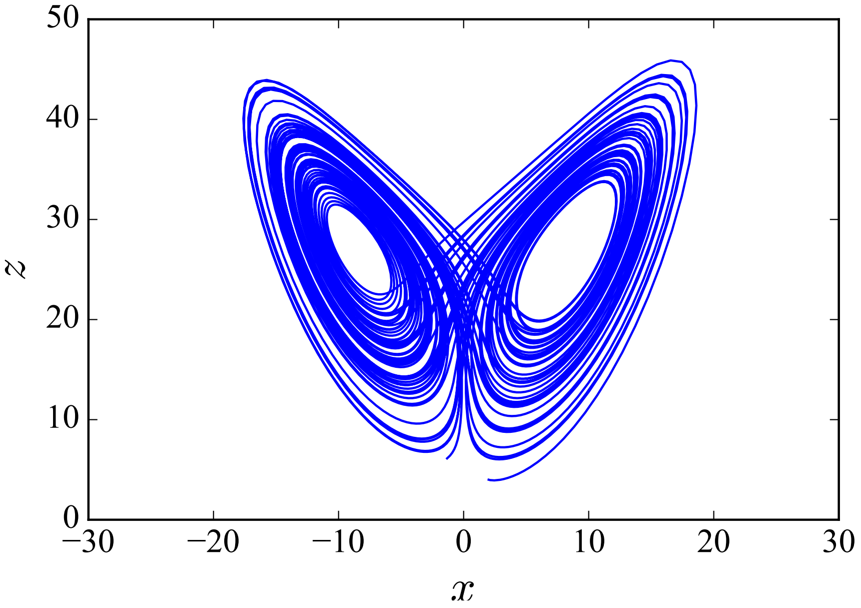


Figure 4.5: Lorenz attractor on xz-plane with $\sigma = 10$, $\beta = 8/3$ and $\rho = 28$ and initial condition is set to be $(x_o, y_o, z_o) = (2, 3, 4)$.

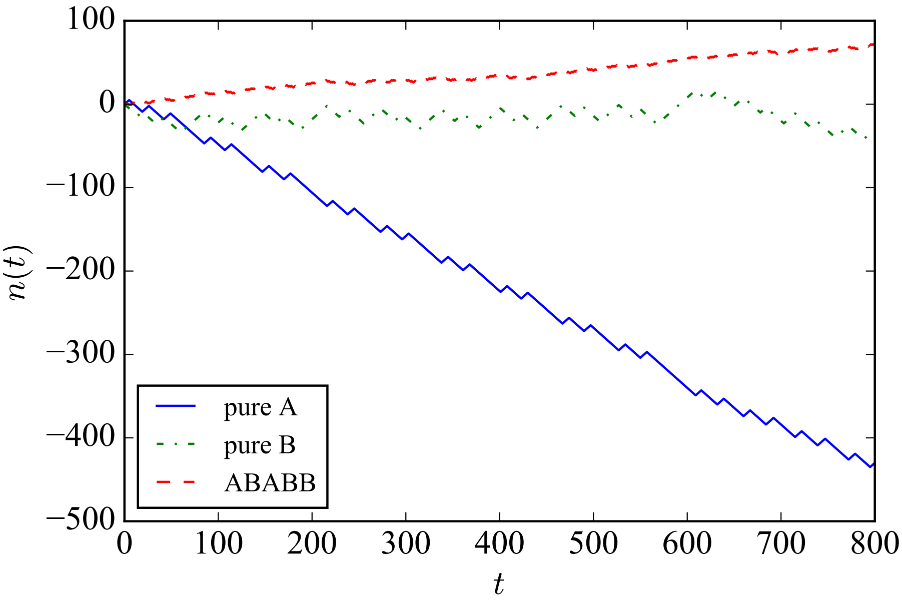


Figure 4.6: Graph of number of tokens n against time with $g_0 = 0.7$ and $g_1 = 0.3$, showing Parrondo effect for Lorenz system with $\rho = 38$. Both pure A sequence (blue) and pure B sequence (green) are losing strategies, while periodic sequence ABABB (red) is a winning strategy.

4.3 Sequential Mixing with Time Warping

Inspired by Parrondo games in the original setting and in toy model of investments, we are interested to see whether such tactic can improve the performance of portfolio involving signal detection and time warping as introduced in Ch.2. Different strategy refers to setting the greed factor to be of different values $g = g_s$. Suppose strategy A corresponds to setting the greed value to be $g = g_0 = 0.1$, strategy B is $g = g_1 = 0.9$ and the reference strategy refers to $g = g_n = 0.5$, we denote the portfolio value evaluated in capital for the three cases above to be V_m^0, V_m^1 and V_m^n respectively. Then, the fractional change of the rate of return for the two strategies with respect to the rate of return for the neutral strategy in 4.19 is obtained.

$$RR_f^s(t) = \frac{V_m^n(t-1)[V_m^s(t) - V_m^s(t-1)] - V_m^s(t-1)[V_m^n(t) - V_m^n(t-1)]}{V_m^s(t-1)[V_m^n(t) - V_m^n(t-1)]}\,, \tag{4.19}$$

where s is either 0 or 1. If $g_s \neq g_{|s-1|}$, then $RR_f^s = RR_f^{|s-1|}$ only when there are no transactions. If $RR_f^s > RR_f^{|s-1|}$, then strategy s should be used since it performs better. On the other hand, if $RR_f^s > RR_f^{|s-1|}$, then strategy $|s-1|$ should be used instead. We refer to the change of greed factor according to the rule above as sequential mixing of investment strategy. In Fig. 4.7 and Fig. 4.8, the time evolution of RR_f is plotted for both strategy A and B.

4.4 Fuzzy Portfolio Management

As shown in Sect.4.3, sequential mixing of investment strategies may improve the performance of a particular portfolio, given that we know in advance which strategy would dominate at a particular instance. Nonetheless, this is not possible when we are making actual investment decisions. If we do know which strategy dominates for a particular transaction, it means we know whether the price will go up or go down, and such investment would involve no risks, which should not occur in an efficient market. Furthermore, the sequence of strategies employed is based on the testing set rather than the training set. Here, we attempt to introduce strategy mixing with a more objective approach.

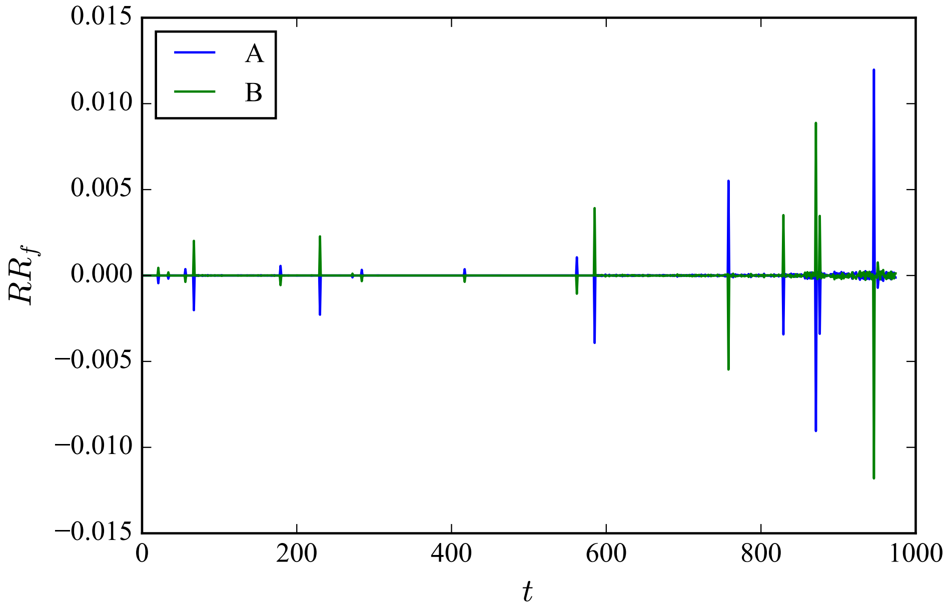


Figure 4.7: The fractional difference of the rate of return of both investment strategies with respect to the reference strategy for HSI. Strategy A (blue) corresponds to $g_0 = 0.1$ and strategy B (green) corresponds to $g_1 = 0.9$.

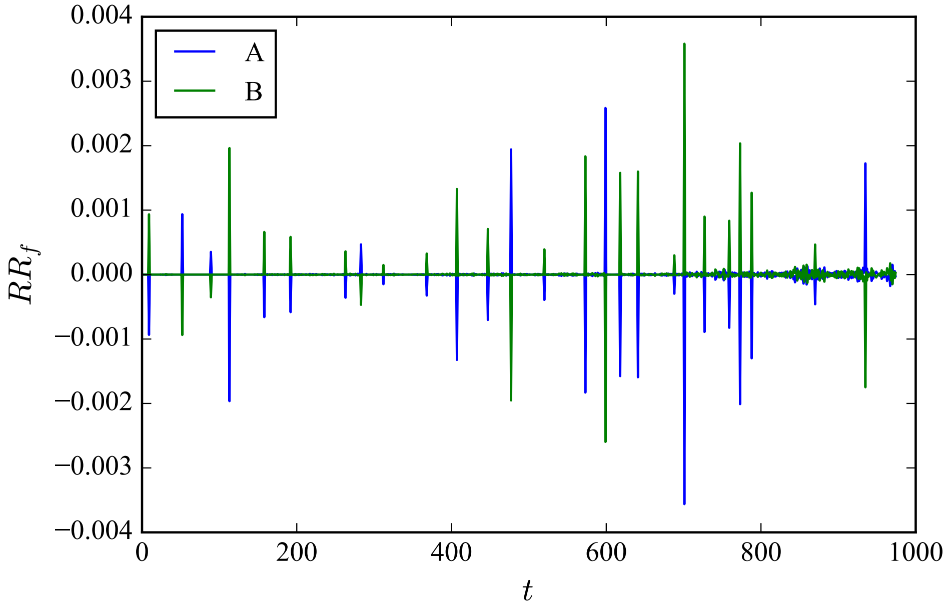


Figure 4.8: The fractional difference of the rate of return of both investment strategies with respect to the reference strategy for FTSE. Strategy A (blue) corresponds to $g_0 = 0.1$ and strategy B (green) corresponds to $g_1 = 0.9$.

4.4.1 Technical Patterns

So far, we are determining whether to buy or to sell based on two observations: chart patterns and rate of return. For chart patterns, we measure the distance between the segment of the time series and the template of technical patterns using time warping. If the distance is smaller than a particular threshold value, we would decide it matches with the pattern. If the distance is larger than the threshold value, then we would determine that particular chart pattern is absent. In another words, we map a continuous variable into binary. This corresponds to the step function in Fig.4.9, which can be referred to as a crisp mapping function. Suppose we define the mapping function to be π_k, we may replace this mapping function with trapezoidal [32, 43]:

$$\pi_k(x) = \begin{cases} 1 & x \leq c_k \\ (d_k - x)/(d_k - c_k) & c_k < x \leq d_k \\ 0 & x > d_k \end{cases} \tag{4.20}$$

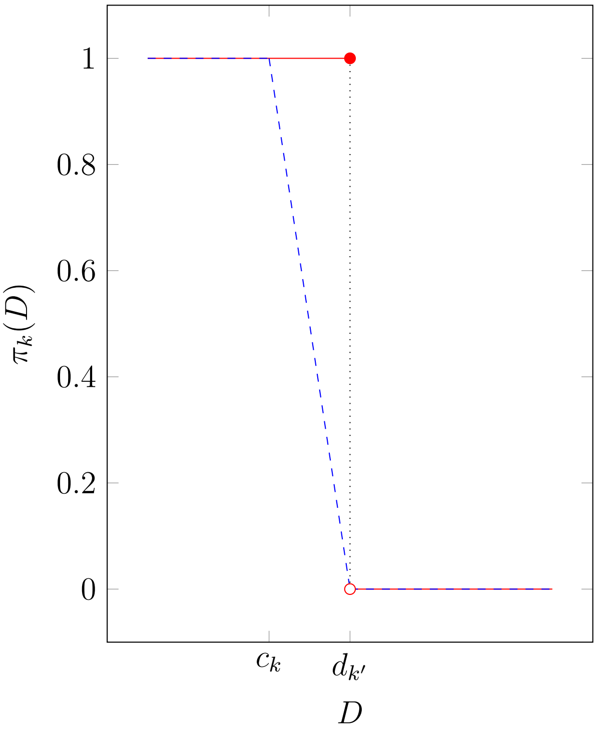


Figure 4.9: Mapping functions for chart pattern defined by step function (red solid line) and trapezoidal rule (blue dotted line).

If the rate of return satisfies the inequality given by the Bayes detector, then instead of using a fixed value of green factor g, we would be use $g = g'\pi_k$, where g' is some constant number such that $g' < 1$. At this point, we would like to keep the instances of transaction to be the same as that in Ch.2, while changing the value of g. Therefore, for practicality, if $D \leq d_0$ at a particular instance, we would not calculate the value of π_1 and assumes it takes the value of 0, which is not true in general.

4.4.2 Rate of Return

For observation of rate of return, we use Bayesian signal detection to determine whether the price might go up or down. However, the evaluating the threshold η involves estimating the costs, namely λ_{ij}. This can be restated in the following manner: Suppose the likelihood ratio is a monotonic decreasing function of x, we may define the critical value of x as x_{crit} such that $\eta = \Lambda(x)$ at $x = x_{crit}$. Therefore, if $x > x_{crit}$, we should take action that correspond to predicting an increase of the stock price since this corresponds to $\eta > \Lambda$. Similarly, if if $x < x_{crit}$, we should expect the stock price to go down and take corresponding action as this is equivalent to $\eta < \Lambda$.

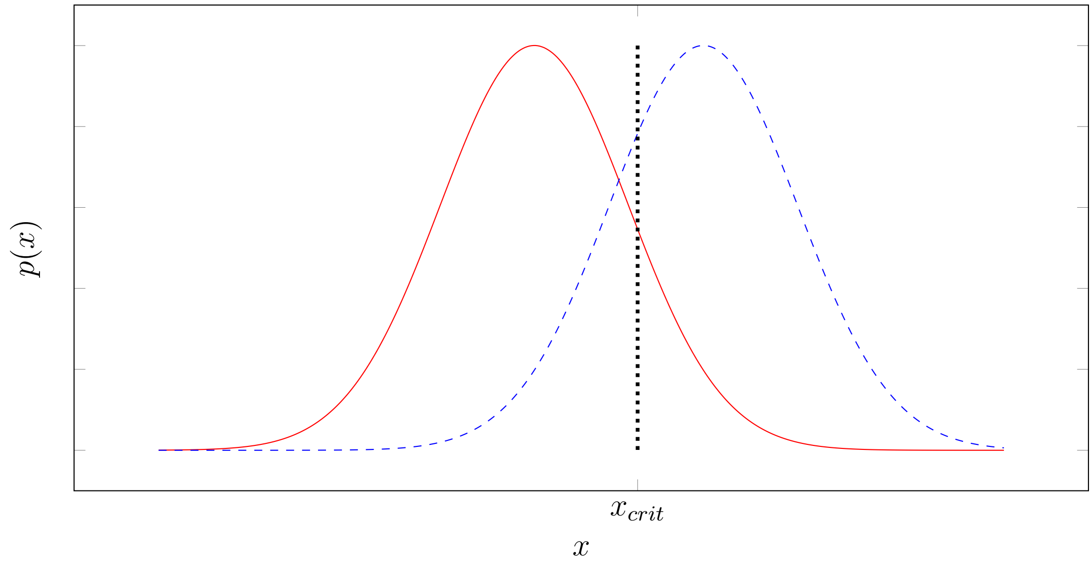


Figure 4.10: Baysian Signal detection using critical value of x, given the probability density functions with the absence of the signal (red solid line) and the presence of the signal (blue dotted line). At $x = x_{crit}$, the likelihood ratio Λ equals to the threshold η.

Different ways of estimating the costs λ_{ij} would lead to different values of thresholds, which in turn changes x_{crit}. Therefore, this would give a distribution of the values of x_{crit}. For simplication, we consider defining the mapping function for up prediction to be ρ_0

mapping function for down prediction as ρ_1.

$$\rho_0 = \begin{cases} 0 & x \leq c_{crit} - c \\ \frac{(x - x_{crit} + c)}{2c} & x_{crit} - c < x \leq x_{crit} + c \\ 1 & x > x_{crit} + c \end{cases}, \tag{4.21}$$

and we define the mapping function for down prediction to be $\rho_1 = 1 - \rho_0$.

$$\rho_1 = \begin{cases} 1 & x \leq x_{crit} - c \\ \frac{(x_{crit} + c - x)}{2c} & x_{crit} - c < x \leq x_{crit} + c \\ 0 & x > x_{crit} + c \end{cases}, \tag{4.22}$$

Here, c controls the "fuzziness" such that when c is very small, the mapping function would be similar to that of a step function. Thus, it is more general compared to the methodology in Ch. 2, which corresponds to the limit as c approaches zero.

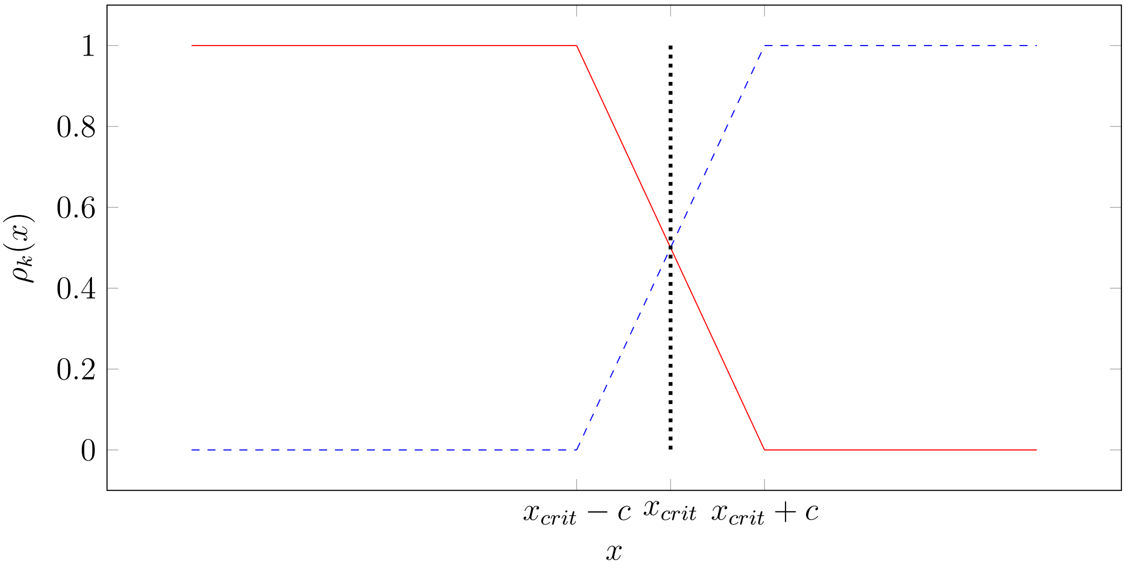


Figure 4.11: Fuzzy mapping functions for Bayes detector, corresponding to hypothesis of U (blue dashed line) and hypothesis of D (green solid line) respectively, while c determines the deviation of acceptable values of x away from x_{crit}.

If we consider crisp classifier of the technical patterns, a.k.a the use of step function for membership function, then given that the condition C_k is satisfied, meaning the segment of the time series matches with the pattern associated to class k of either U or D, the greed factor is set to be $g = g'\rho_k$.

4.4.3 General Case

Suppose now that condition C_k is satisfied, which indicates the distance obtained by time warping of the pattern corresponding to k is smaller than d_k, then one may consider modifying the greed factor as

$$g(D,x) = g' \min(\pi_k(D)\rho_k(x)) \,. \tag{4.23}$$

The reason that g' is usually set to be smaller than 1 is to avoid selling all the shares possessed or using all capitals to purchase stocks. In other words, g' is the maximum value of the greed factor g.

4.4.4 Implementation

We first focus on π_k that takes the form of trapezoidal rule as defined in Eq. (4.20), while ρ_k is defined by step function. Furthermore, we use the same variables as previous section, with (Π_0, d_0) to be (BTOPD,0.023) and (Π_1, d_1) to be (SD, 0.02) respectively. In contrast to the previous chapter, however, we need to determine (c_0, c_1) instead of g. Here, we would like to discuss the range of fuzzy set parameters in portfolio optimization. For simplicity, we assume $d_0 - c_0 = d_1 - c_1 = c$ and c is much smaller than d_0 and d_1. Notice that under this scheme, the number of predicted U and D would be the same as the scheme in Sect. 3.5, since for $\pi_k(D) > 0$ for $D < d_k$. All portfolios would therefore involve transactions at the same instances regardless of the value of c, while the amount of capital or shares involved would be different. At $c = 0$, the portfolio constructed would be equivalent to the portfolio in Sect. 2, giving the same PV. As c approaches $-\infty$, average greed factor $\langle g \rangle$ also approaches 0 and $PV_f = 0$. So, one would expect that if the chart patterns are good prediction rules that gives perfect prediction or the number of transaction is large, the portfolio value for $d > c > 0$ would fall between the two limits and the performance of the portfolio would be poorer as c increases, as larger values of c effectively reduce the average value of g. In reality, PV_f can be negative for certain periods of time.

In contrast, if we use step function for the mapping function of the chart patterns π_k, and use Eq. (4.21), (4.22) for mapping function ρ_k, the number of transactions would be different. In particular, for larger c, the number of transaction would also increase as input signals that are originally rejected by Baysian detector would be accepted with a greed

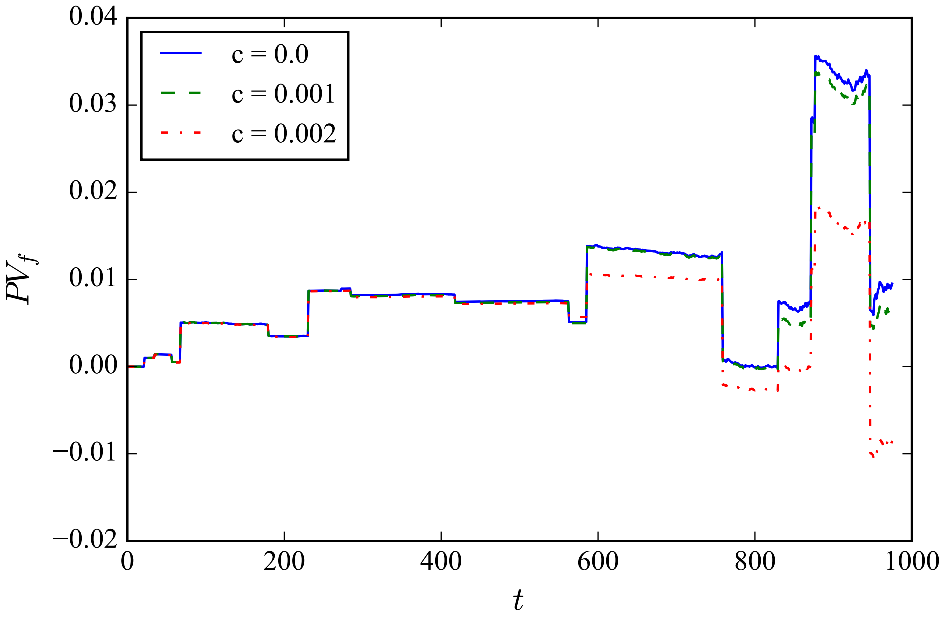


Figure 4.12: Fractional change of portfolio value is plotted against time for $(\Pi_0, d_0) = (BTOPD, 0.023)$ and $(\Pi_1, d_1) = (SD, 0.02)$ for HSI with fuzzy mapping function for chart pattern and crisp mapping function for signal detection with rate of return.

factor g that is smaller than 0.5. Nonetheless, the investor would also have less confidence when the rate of return x is just slightly larger than x_{crit}, reflected by a smaller value of g. Because using different values of c for ρ_k would affect the number of transactions, we only consider small values of c in portfolio optimization. In this section, c falls in the range from 10^{-4} to 10^{-3}. In contrast, we adopt larger values of c for π_k in portfolio optimization since it does not affect the number of transactions. In this section, we consider the range from 10^{-3} to 10^{-2}.

An interesting feature is that as c increases, there is a sudden change of portfolio value that drops abruptly between $c = 0.0009$ and $c = 0.0012$. This is due to the change of number of transactions and due to the change of confidence, as mentioned above. The value of c has to be larger than some critical value in order to accept a particular hypothesis or speculation of the market and to perform manipulation of their portfolio composition for x that deviates from x_{crit}, which may then lead to large changes of the portfolio value. For example, one can observe from Fig. 4.13that the transaction around $t = 250$ is the cause of such leaps or differences in terms of performances. On the other hand, similar to the previous case, the portfolio value in general decreases with increasing c on a large scale due to such abrupt changes. But if one focus on large t, the portfolio

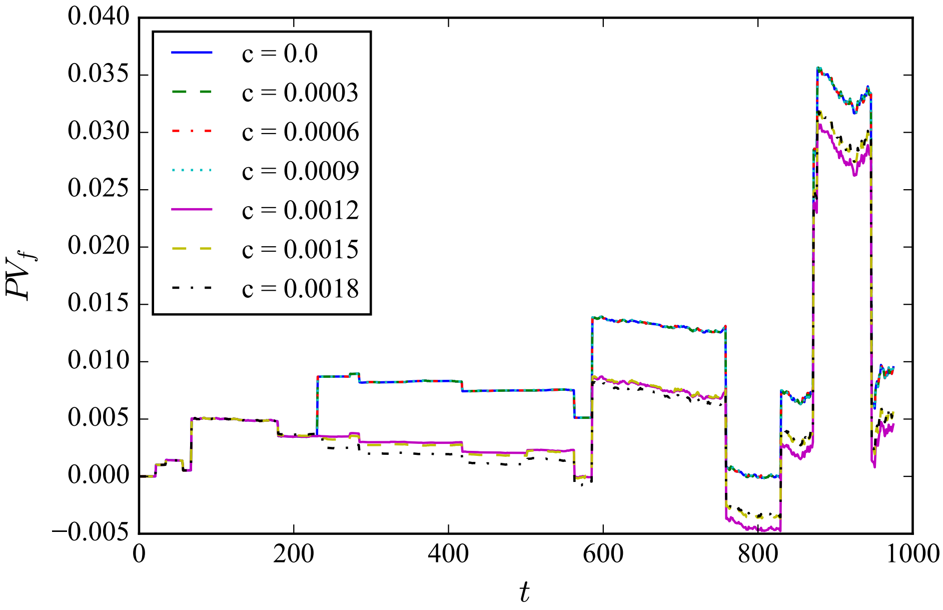


Figure 4.13: Fractional change of portfolio value is plotted against time for $(\Pi_0, d_0) = (BTOPD, 0.023)$ and $(\Pi_1, d_1) = (SD, 0.02)$for HSI with crisp mapping function for chart pattern and fuzzy mapping function for signal detection with rate of return.

actually increases with increasing c for $c \geq 0.0012$.

So far, we have selected rules in the form of chart patterns that gives good predictions of future pricing. From Fig. 3.7, one observes that BTOPD has a smaller e when d is less than 0.023, whereas e is decreasing for SD with smaller values of d, and has a second minimal at 0.023, which should give a poorer performance based on the results from the training set. Thus, we use a crisp mapping function for π_0 with the same $d_0 = 0.023$ as before and use a fuzzy mapping function for π_1 with a slightly larger $d_1 = 0.023$ compared to previous settings, and we try to see if fuzzification can leads to better portfolio performances. Same as before, we are first concerned with crisp definition for ρ_k and π_0, and fuzzy mapping function for π_1, and plot out the fractional difference for portfolio values compared to fluctuation of the index in Fig. 4.14. As predicted, while the best-performing portfolios still attain a value that is less the portfolio with $d_1 = 0.02$ and $c = 0$, constructed portfolios with fuzzy mapping function for chart patterns out performs the portfolios with crisp mapping function for certain values of c that are non-zero under the current prediction rules. Such improvement can be observed early on since $t < 100$. And obviously, using different sets of prediction rules would give different estimations of the average cost or the threshold, which in turn would give different values of x_{crit}. There

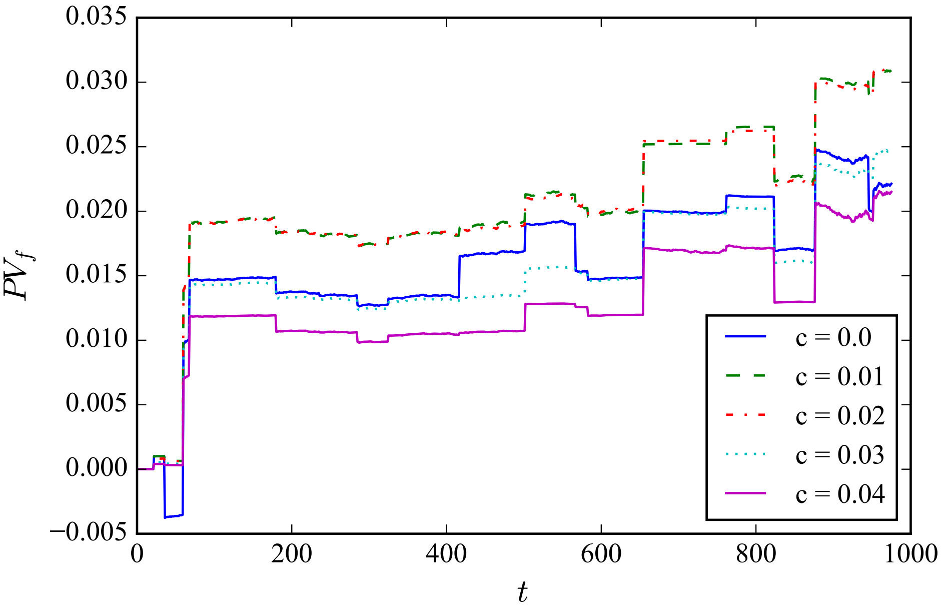


Figure 4.14: Fractional change of portfolio value is plotted against time for $(\Pi_0, d_0) = (BTOPD, 0.023)$ and $(\Pi_1, d_1) = (SD, 0.023)$ for HSI with fuzzy mapping function for π_1 and crisp mapping function for ρ_k and π_0.

are in total 8 predictions for D and 10 predictions for U.

Next, we are interested in keeping both π_0 and π_1 as crisp mapping functions, while using fuzzy mapping functions for ρ_0 and ρ_1 with $c = d_0 - c_0 = d_1 - c_1$. The value of c is incremented from 0 to 0.002 with step size of 0.0003 to observe how the portfolio performs. The fractional portfolio values are plotted in Fig. 4.15.

Again, fuzzy portfolio management shows better performance at certain non-zero values of c compared to traditional portfolio management. This time, the separation of trajectories of PV_f for different c can be observed after $t = 500$, while there are no large differences for smaller t in contrast to the previous case. This encourages us to combine both cases in constructing a portfolio. From Fig. 4.15, the best performing portfolio corresponds to $c = 0.0018$. Hence, for ρ_k, we set $c = 0.0018$ while vary the c parameter for π_1. Same as before, we still use step function to define π_0, or equivalently, set $c = 0$. It is noted that Parrondo effect is again observed, as $c = 0$ does not give the best portfolio performance. Through fuzzy management of portfolio, we have introduced sequential mixing of different investment strategies defined by greed factor $g(t)$, thus leading to a higher portfolio value compared to pure strategies where g is constant.

Similar to previous section, we also consider using the closing price of FTSE. Using the

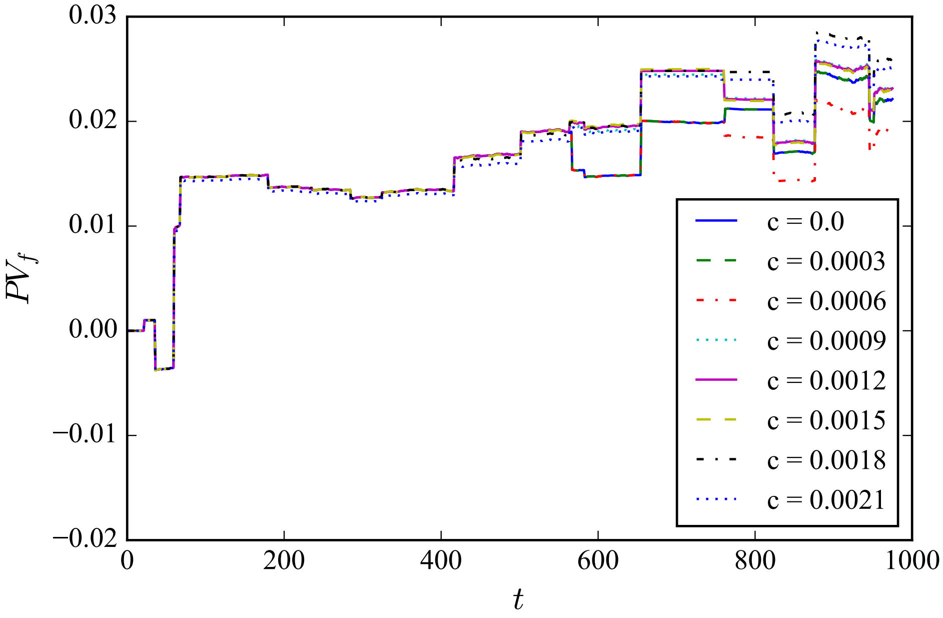


Figure 4.15: Fractional change of portfolio value is plotted against time for $(\Pi_0, d_0) = (BTOPD, 0.023)$ and $(\Pi_1, d_1) = (SD, 0.023)$ for HSI with fuzzy mapping function for ρ_k and crisp mapping function for π_k with $k = 0, 1$.

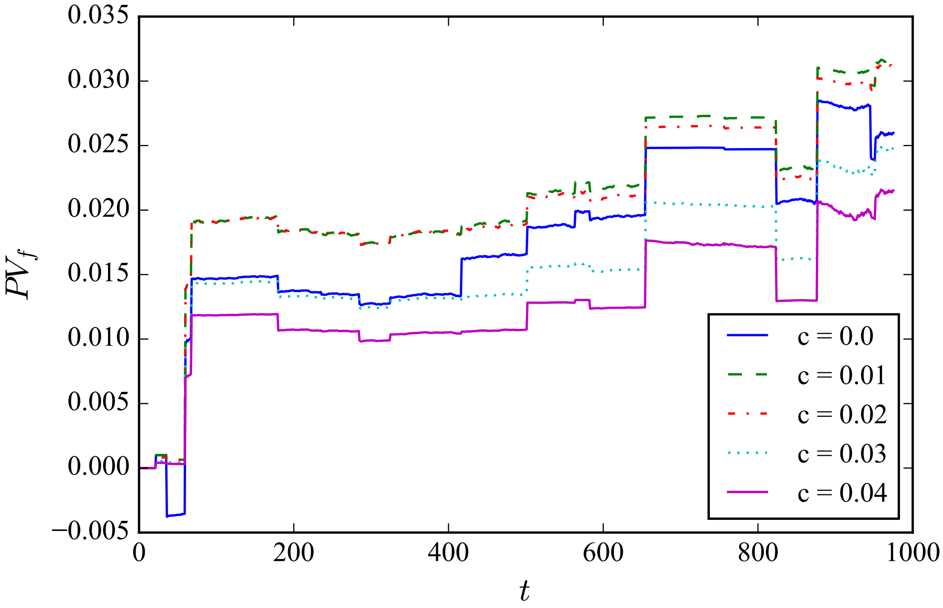


Figure 4.16: Fractional change of portfolio value is plotted against time for $(\Pi_0, d_0) = (BTOPD, 0.023)$ and $(\Pi_1, d_1) = (SD, 0.023)$ for HSI with fuzzy mapping function for ρ_k with $c = 0.0018$ and π_1 with varying c and crisp mapping function for π_0.

same patterns as before, which are $(\Pi_0, d_0) = (RB, 0.023)$, and $(\Pi_1, d_1) = (SU, 0.029)$, we fuzzify the mapping functions for patterns only and show the portfolio value in Fig. 4.17, which shows that portfolio with small c would perform better. This is similar to what was observed in Fig. 4.12. We then shifted the values of d_0 to 0.024 to obtain Fig. 4.18. Similar to previous case in Fig. 4.14, we would fuzzify the mapping function of one of the two patterns only. Since there are more U predictions than D predictions for FTSE, we would fuzzify π_0 and use crisp function for π_1. Since the shift of d is small, we only vary c from 0 to 0.0008 with 0.0002-step increment. Over this period, the portfolio with $c = 0.0006$ performs better than the others. Next, we use fuzzy mapping function for rate of return π_k, while both π_0 and π_1 are step functions. During this period, the portfolio performs the best at $c = 0$. However, the portfolio value is not monotonic decreasing with increasing c as shown in Fig.4.19.

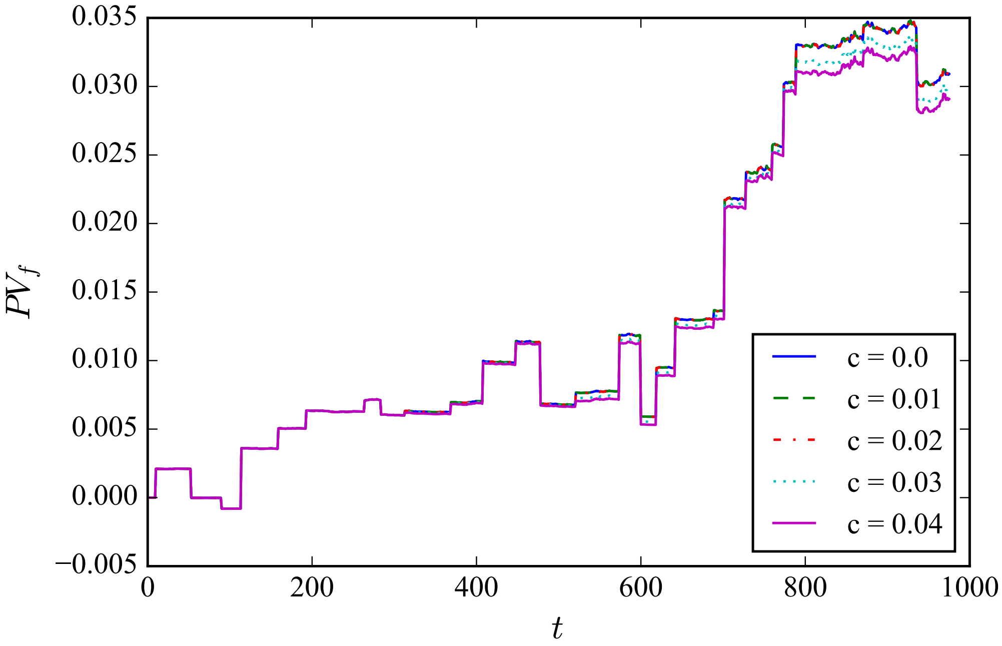


Figure 4.17: Fractional change of portfolio value is plotted against time for $(\Pi_0, d_0) = (RB, 0.023)$ and $(\Pi_1, d_1) = (SU, 0.029)$ for FTSE with fuzzy mapping function for chart pattern and crisp mapping function for signal detection with rate of return.

4.5 Summary

In this section, Parrondo games is linked up with portfolio management. Previously, we show that some investment strategies performs better than others during different periods, and we discuss here the sequential mixing of investment strategies in boosting

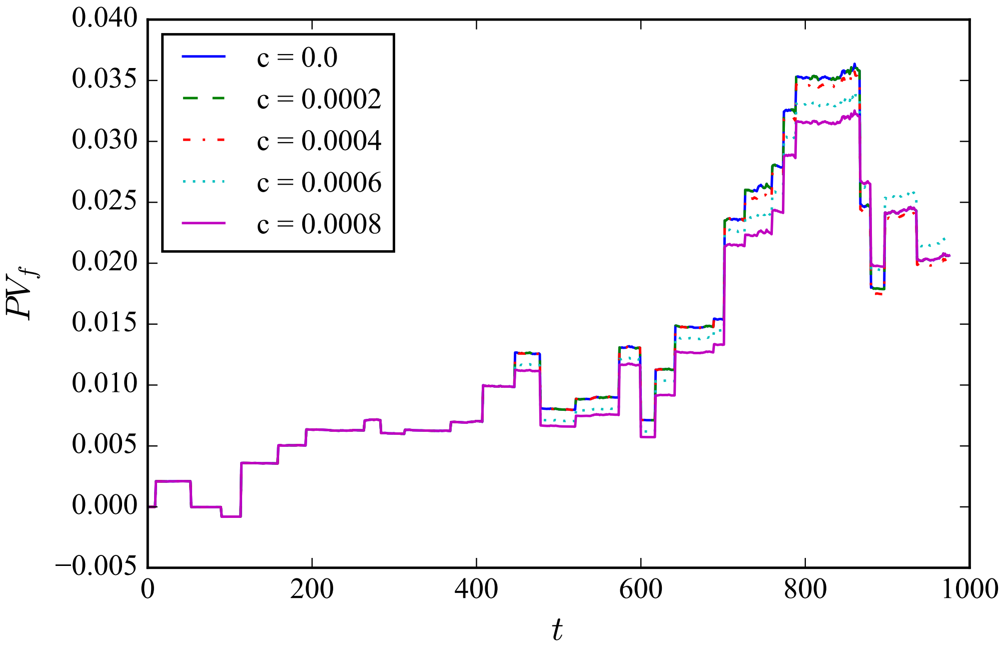


Figure 4.18: Fractional change of portfolio value is plotted against time for $(\Pi_0, d_0) = (RB, 0.024)$ and $(\Pi_1, d_1) = (SU, 0.029)$ for FTSE with fuzzy mapping function for π_0 and crisp mapping function for ρ_k and π_1.

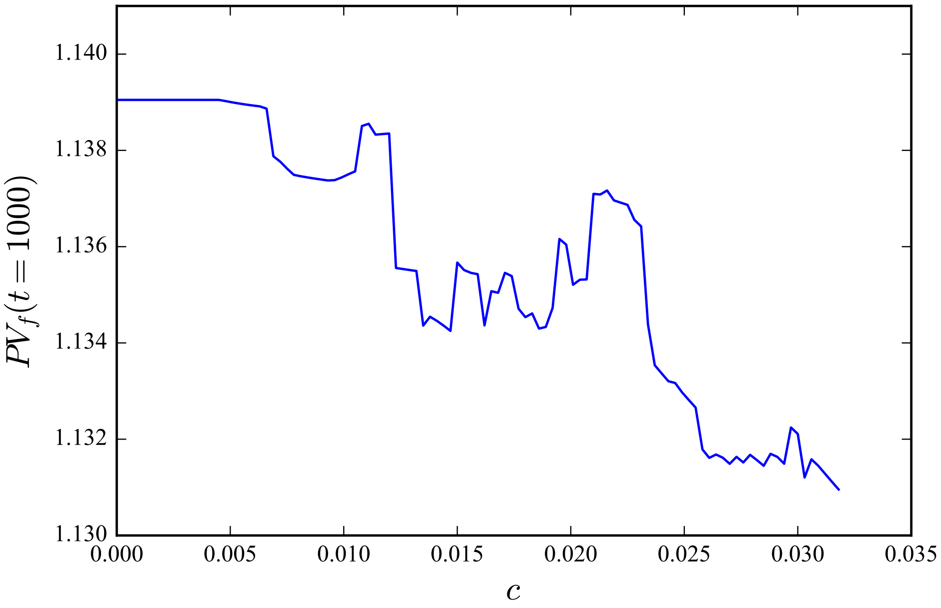


Figure 4.19: Fractional change of portfolio value at $t = 1000$ is plotted against c for $(\Pi_0, d_0) = (RB, 0.023)$ and $(\Pi_1, d_1) = (SU, 0.029)$ for FTSE with fuzzy mapping function for π_0 and crisp mapping function for ρ_k and π_1.

the portfolio performance. In particular, fuzzy portfolio management is proposed, and is shown to perform better in some cases for real data at some values of c, similar to Parrondo effect in the toy model with artificial time series.

Chapter 5

Conclusion

5.1 Summary

In this thesis, we have investigated the application of time warping techniques on financial time series. In the first chapter, we gave an introduction to this thesis. In the second chapter, we introduce the concept of time warping, and illustrate two methodologies, which are based on genetic algorithm and dynamic programming respectively. For the GA-based approach, we introduce the Discrete Time Warping Genetic Algorithm, which relies on mutation for the diversity in the gene pool and does not involve crossover operators. We further demonstrate the use of dTWGA on constructing the description of the financial network using the closing prices of the index components. For DP-based approach, we review the procedures of dynamic time warping, which is used in the subsequent sections for portfolio management.

In the third chapter, we apply DTW on portfolio management with cash and one risky asset. We first review some of the results in Baysian Signal Detection Theory and propose the use of SDT on the daily rate of return of a stock. We then give the definitions of the chart patterns, which can be categorized into trend reversal patterns and trend continuation patterns. Applying DTW to recognize the chart patterns in the time series of the stock price allow us to employ a rule-based approach for trading. We use SDT in combination with DTW for trading and conduct the experiment on both Hang Seng Index (HSI) and Financial Times Stock Exchange 100 index (FTSE). In addition to historic prices, we also considered the chart patterns in historic data of volume. Apart from predefined patterns, we use k-means algorithm to extract potential chart patterns and apply them on portfolio management.

In the last chapter, we try the relate portfolio management with Parrondo games. We review the settings for the original Parrondo games, in which the states are defined by the capital possessed by the player. Then, we introduce a toy model which to demonstrate Parrondo effect in the context of portfolio management with artificial time series. Parrondo effect is demonstrated for both sinusoidal and Lorenz system. Next, we discuss the possibility of mixing the greed factors or trading strategies in a sequential manner in order to improve the performance of a portfolio, similar to the mixing of two games to produce Parrondo effect. Lastly, we link up the methodology for portfolio management proposed in Ch. 2 with fuzzy set as a way to conduct sequential mixing. We have experimented on the fuzzification for both the technical pattern and the rate of return on HSI and FTSE, and shows that Fuzzy Portfolio Management can sometimes perform better than the original portfolio.

5.2 Future Work

In this thesis, we proposed the use of time warping in pattern recognition in financial time series. DTW or dTWGA can be applied to detect or measure the coupling between different time series, particular for the data with different sampling rates or for time series with different lengths, and might be useful for studies involving climatic and biological time series. In the context of portfolio management, it would be interesting to study the control-feedback mechanism. For example, one may be intrigued in the emergence of the chart patterns in both the historic price and in volume data when the information of the prices and volumes are available globally to all investors. Another issue of interest is to place the investors in social networks defined by different topologies, in which the chartists are allowed to communicate. Although the return yielded by our method is lower than existing trading schemes, the magnitude of the return is reasonable since we only consider the prediction of a single event of either U or D for each trading. For improvements and developments of algorithms for trading, one may want to incorporate technical patterns and time warping in designing momentum indicators.

Bibliography

[1] Rosario N. Mantegna. Hierarchical structure in financial markets. *The European Physical Journal B-Condensed Matter and Complex Systems*, 11(1):193–197, 1999.

[2] J.-P. Onnela, Anirban Chakraborti, Kimmo Kaski, and Janos Kertesz. Dynamic asset trees and Black Monday. *Physica A: Statistical Mechanics and its Applications*, 324(1):247–252, 2003.

[3] Leonidas Sandoval and Italo De Paula Franca. Correlation of financial markets in times of crisis. *Physica A: Statistical Mechanics and its Applications*, 391(1):187–208, 2012.

[4] K. Tse Chi, Jing Liu, and Francis CM Lau. A network perspective of the stock market. *Journal of Empirical Finance*, 17(4):659–667, 2010.

[5] Young-Il Moon, Balaji Rajagopalan, and Upmanu Lall. Estimation of mutual information using kernel density estimators. *Physical Review E*, 52(3):2318, 1995.

[6] Alexander Kraskov, Harald Stgbauer, and Peter Grassberger. Estimating mutual information. *Physical review E*, 69(6):066138, 2004.

[7] Greg Ver Steeg and Aram Galstyan. Information transfer in social media. In *Proceedings of the 21st international conference on World Wide Web*, pages 509–518. ACM, 2012.

[8] Jakob Runge, Jobst Heitzig, Vladimir Petoukhov, and Jrgen Kurths. Escaping the curse of dimensionality in estimating multivariate transfer entropy. *Physical review letters*, 108(25):258701, 2012.

[9] Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing*, 26(1):43–49, 1978.

[10] Prodromos Tsinaslanidis, Antonis Alexandridis, Achilleas Zapranis, and Efstratios Livanis. Dynamic time warping as a similarity measure: applications in finance.

[11] Joseph Di Martino. Dynamic time warping algorithms for isolated and connected word recognition. In *New systems and architectures for automatic speech recognition and synthesis*, pages 405–418. Springer, 1985.

[12] Volkan Tuzcu and Selman Nas. Dynamic time warping as a novel tool in pattern recognition of ECG changes in heart rhythm disturbances. In *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, volume 1, pages 182–186. IEEE, 2005.

[13] B. S. Raghavendra, Deep Bera, Ajit S. Bopardikar, and Rangavittal Narayanan. Cardiac arrhythmia detection using dynamic time warping of ECG beats in e-healthcare systems. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2011 IEEE International Symposium on a*, pages 1–6. IEEE, 2011.

[14] Chotirat Ann Ratanamahatana and Eamonn Keogh. Making time-series classification more accurate using learned constraints. In *Proceedings of the 2004 SIAM International Conference on Data Mining*, pages 11–22. SIAM, 2004.

[15] Ilan D. Shallom, Raziel Haimi-Cohen, and Tal Golan. Dynamic time warping with boundaries constraint relaxation. In *Electrical and Electronics Engineers in Israel, 1989. The Sixteenth Conference of*, pages 1–4. IEEE, 1989.

[16] Pankaj Kumar, Ankur Gupta, Valadi K. Jayaraman, and BhaskarD Kulkarni. Aligning time series with genetically tuned dynamic time warping algorithm. *Advances in Metaheuristics for Hard Optimization*, pages 251–261, 2008.

[17] Slimane Sefiane and Mohamed Benbouziane. Portfolio selection using genetic algorithm. *Journal of Applied Finance and Banking*, 2(4):143, 2012.

[18] King Loong Shiu and Kwok Yip Szeto. Self-adaptive mutation only genetic algorithm: an application on the optimization of airport capacity utilization. In *International*

Conference on Intelligent Data Engineering and Automated Learning, pages 428–435. Springer, 2008.

[19] Kwok Yip Szeto and Jian Zhang. Adaptive genetic algorithm and quasi-parallel genetic algorithm: Application to knapsack problem. In *International Conference on Large-Scale Scientific Computing*, pages 189–196. Springer, 2005.

[20] Travis S. Metcalfe and Paul Charbonneau. Stellar structure modeling using a parallel genetic algorithm for objective global optimization. *Journal of Computational Physics*, 185(1):176–193, 2003.

[21] Patrick Wan-Hin Luk, Ga Ching Lui, and Kwok Yip Szeto. Optimization of systemic stability of directed network using genetic algorithm. In *Computer and Information Science (ICIS), 2016 IEEE/ACIS 15th International Conference on*, pages 1–6. IEEE, 2016.

[22] Wenshuo Guo and Kwok Yip Szeto. Minimization of Systemic Risk for Directed Network Using Genetic Algorithm. In *European Conference on the Applications of Evolutionary Computation*, pages 3–16. Springer, 2017.

[23] Yi-Kuei Lin and Cheng-Ta Yeh. Maximal network reliability with optimal transmission line assignment for stochastic electric power networks via genetic algorithms. *Applied Soft Computing*, 11(2):2714–2724, 2011.

[24] Rodger Ziemer and William H. Tranter. *Principles Of Communications: System Modulation And Noise*. John Wiley & Sons, 2006.

[25] J. M. R. Parrondo. How to cheat a bad mathematician. *EEC HC&M Network on Complexity and Chaos*, 1996.

[26] Gregory P. Harmer and Derek Abbott. Game theory: Losing strategies can win by Parrondo's paradox. *Nature*, 402(6764):864–864, 1999.

[27] Greg P. Harmer and Derek Abbott. Parrondo's paradox. *Statistical Science*, pages 206–213, 1999.

[28] Luis Dinis. Optimal sequence for Parrondo games. *Physical Review E*, 77(2):021124, 2008.

[29] Ka Wai Cheung, Ho Fai Ma, Degang Wu, Ga Ching Lui, and Kwok Yip Szeto. Winning in sequential Parrondo games by players with short-term memory. *Journal of Statistical Mechanics: Theory and Experiment*, 2016(5):054042, 2016.

[30] Degang Wu and Kwok Yip Szeto. Extended Parrondo's game and Brownian ratchets: Strong and weak Parrondo effect. *Physical Review E*, 89(2):022142, 2014.

[31] Lotfi A. Zadeh. Fuzzy sets. *Information and control*, 8(3):338–353, 1965.

[32] George J. Klir, Bo Yuan, and Ute H. St Clair. *Fuzzy set theory: foundations and applications.* 1997.

[33] Christian Wagner and Hani Hagras. Toward general type-2 fuzzy logic systems based on zSlices. *IEEE Transactions on Fuzzy Systems*, 18(4):637–660, 2010.

[34] Richard Bellman. On the theory of dynamic programming. *Proceedings of the National Academy of Sciences*, 38(8):716–719, 1952.

[35] R. Bellman and E. Lee. History and development of dynamic programming. *IEEE Control Systems Magazine*, 4(4):24–28, 1984.

[36] Richard Ernest Bellman. An introduction to the theory of dynamic programming. 1953.

[37] Thomas H. Cormen. *Introduction to algorithms.* MIT press, 2009.

[38] John J. Murphy. *Technical analysis of the financial markets: A comprehensive guide to trading methods and applications.* Penguin, 1999.

[39] Andrew W. Lo, Harry Mamaysky, and Jiang Wang. Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation. *The journal of finance*, 55(4):1705–1770, 2000.

[40] Tak-chung Fu, Fu-lai Chung, Robert Luk, and Chak-man Ng. Stock time series pattern matching: Template-based vs. rule-based approaches. *Engineering Applications of Artificial Intelligence*, 20(3):347–364, 2007.

[41] Pang-Ning Tan and others. *Introduction to data mining.* Pearson Education India, 2006.

[42] Alex Smola and S. V. N. Vishwanathan. Introduction to machine learning. *Cambridge University, UK*, 32:34, 2008.

[43] Michael Negnevitsky. *Artificial intelligence: a guide to intelligent systems*. Pearson Education, 2005.

List of Publications

1. Ma, H. F., Cheung, K. W., **Lui, G. C.**, Wu, D., & Szeto, K. Y. (2017). Effect of information exchange in a social network on investment. Computational Economics, 1-13.

2. **Lui, G. C.**, Yip, C. Y., & Szeto, K. Y. (2017, June). Topological evolution of financial network: a genetic algorithmic approach. In International Conference on Hybrid Artificial Intelligence Systems (pp. 113-124). Springer, Cham.

3. **Lui, G. C.**, Wu, D., Cheung, K. W., Ma, H. F., & Szeto, K. Y. (2016, July). Time warping of apneic ECG signals using genetic algorithm. In Evolutionary Computation (CEC), 2016 IEEE Congress on (pp. 178-184). IEEE.

4. Luk, P. W. H., **Lui, G. C.**, & Szeto, K. Y. (2016, June). Optimization of systemic stability of directed network using genetic algorithm. In Computer and Information Science (ICIS), 2016 IEEE/ACIS 15th International Conference on (pp. 1-6). IEEE.

5. Cheung, K. W., Ma, H. F., Wu, D., **Lui, G. C.**, & Szeto, K. Y. (2016). Winning in sequential Parrondo games by players with short-term memory. Journal of Statistical Mechanics: Theory and Experiment, 2016(5), 054042.

Drafts and Publications

Time Warping of Apenic ECG Signals using Genetic Algorithm

Ga Ching Lui, Degang Wu, Ka Wai Cheung, Ho Fai Ma, Kwok Yip Szeto

Abstract

We construct a method of time warping in quasiperiodic time series analysis using genetic algorithm in order to extract the instantaneous phase difference between a template signal and a testing signal. Contrast to previous studies, which involves correlation estimations to determine the shape similarity of two signals taken from the quasiperiodic time series, time warping perform the comparison of the two signals by first constructing a discrete set of M points formed from uniformly sampled values of the template signal f(t). The discrete set of sample values of the testing signal, g(t′), which contains N points, will be interpolated to form a continuous function so that the difference between the template signal at those M points and the corresponding testing signals are minimize to best preserve the mapping of the two signals. The result of this optimization procedure produces a phase shift function that relates the time t′ in the testing signal to the time t in the template signal. Due to the numerous choices in the partitioning of the time domain of the two signals, genetic algorithm is found to be effective in extracting this phase shift function. We apply this theoretical tool of time warping using genetic algorithm to analyze the electrocardiographic (ECG) signals, with the aim to investigate if central apneic and obstructive apneic episodes can be differentiated from non-apneic episodes. Detailed statistical analysis of the phase shift from real ECG data of sleep apnea patient indicates that the difference of both magnitude and phase of the signals can be used to differentiate apneic events from non-apneic events.

I. INTRODUCTION

The analysis of quasiperiodic time series is of great importance in a variety of fields such as biomedical science [1–3] , image processing[4], climatology[5] and finance[6, 7]. Frequency-domain algorithms have been extensively developed for the analysis of quasi-periodic signals, with wavelet transform[8] and Hilbert-Huang transform[9] being the prime examples that are frequently utilized by researchers. An alternative method is to perform the analysis of the time series in the time-domain. In this regard, Dynamic Time Warping (DTW)[10], for example, is a similarity measure[6, 11] which accommodates signals that are similar, but locally out of phase, by using non-linear alignments between two time series. This method has often been used as a similarity measure to classify time series [12], and is quite useful in the study of the relative phase relations between two time series similar in shape[4]. According to Syeda-Mahmood, Beymer and Wang [12], the ECG records of patients having the same type of cardiovascular diseases are often similar in shape, but often out of phase. Therefore DTW becomes an important tool for the study of sleep apnea. However, the performance of DTW depends on the many parameters that are often counter-intuitively small and problem dependent [13], thereby limiting its use in practice. In order to overcome this limitation of DTW, we propose in this paper a genetic algorithm to perform the nonlinear alignment and we call this method time warping using genetic algorithm or TWGA.

The successful application of genetic algorithm (GA) has been demonstrated in fields such as biology[14–16] , clusters [17–19] and glass transition [20] in condensed matter physics. In engineering, genetic algorithm has also been used with good results in problems such as cyclic-steam oil production optimization problem [21], speed control of brushless DC motor [22], airport scheduling [23], mobile robot motion control [24], modeling adaptive agents in stock markets[25, 26], portfolio management[27] and traveling salesman problem[28]. GA has also been applied to optimize the parameters for DTW[29].

Different variants of DTW algorithms have been applied in the detection and classification of abnormal episodes in ECG records that may or may not belong to patients with cardiovascular diseases [12, 30–32], but mostly has one focused on the residual error[12], which is the difference in signal magnitude after alignment, and rarely has studied the phase shifts in ECG records. In this work we introduce a Time Warping Genetic Algorithm (TWGA), with application in the investigation of instantaneous phase differences between

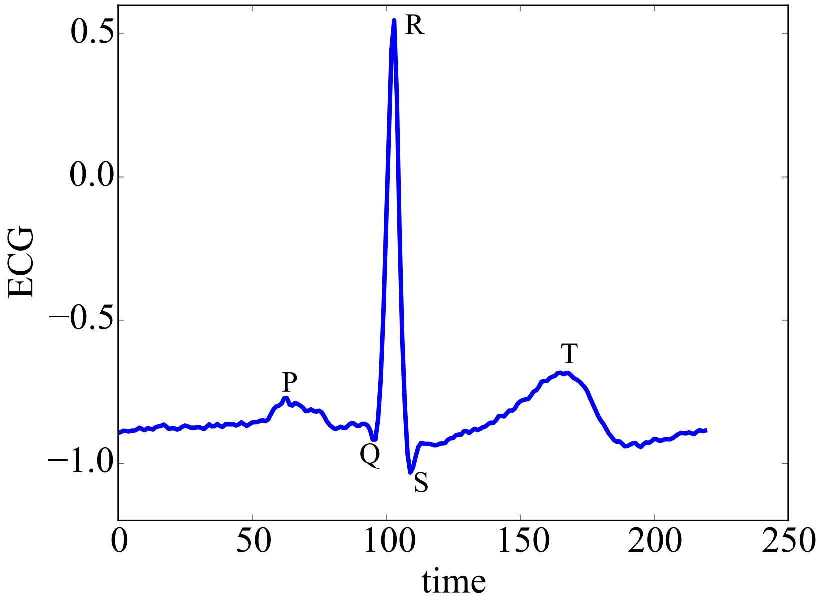


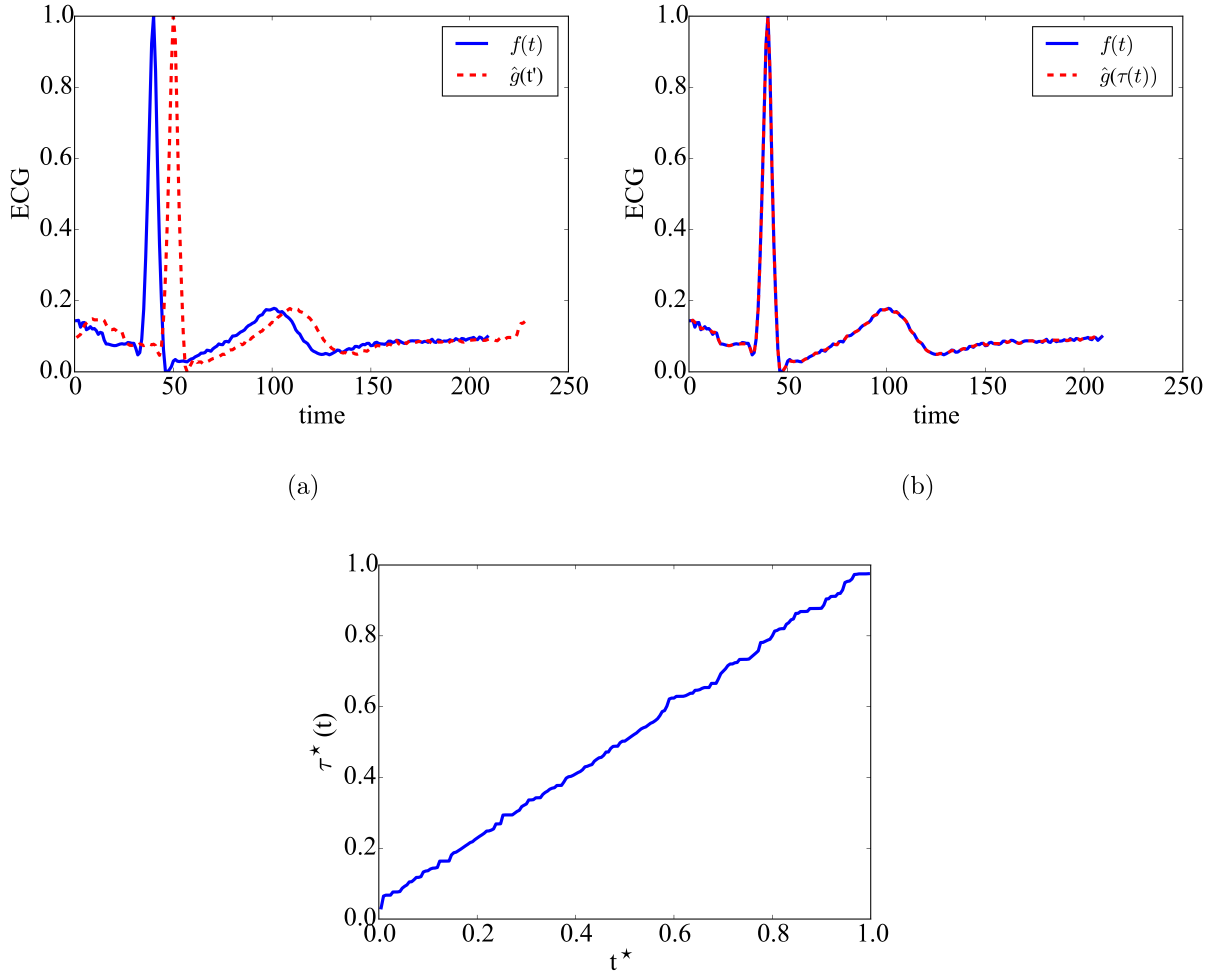
FIG. 1: An example of a quasi-period of ECG signal extracted from slp60 record . P and T denote the P-wave and T-wave. QRS denotes the QRS-complex.

two electrocardiographic signal (ECG) signals for sleep apnea patients.

The time series of ECG signals consist of sequence of pulses. For an electrocardiographic signal (ECG), a quasi-period, or a cardiac cycle, as shown in FIG.1, is comprised of a p-wave, a QRS complex and a t-wave corresponding to atrial depolarization, ventricular depolarization and ventricular repolarization respectively[12, 33].

As a demonstration, an example is given in FIG.2 showing the mapping function between two consecutive quasi-periods. In this example, the boundaries are defined to be the peak positions of the P-waves. The ECG data are obtained from MIT-BIH Polysomnographic Database[34] which can be accessed through `http://physionet.org`[35]. We mainly focus on the record `slp60`. Position of the P-waves can be extracted with ecgpuwave from PhysioToolkit library[35]. The sampling rate for ECG is 250Hz and the sleeping stage is recorded in the annotation files every 30 seconds. We will focus on the normal episodes(NE), central apneic (CA) and obstructive apneic (OA) episodes during sleep stage 1 of `slp60`.

The paper is organized into the several sections. We first define the problem of shape similarity measure in section II. We then introduce Time Warping Genetic Algorithm (TWGA) in section III and apply this method to ECG signals in section IV. The results along with detailed statistical analysis and hypothesis testing are in section V. We end the paper with a discussion of the possible application and future research of TWGA to medical time series as well as other fields in the last section.



(a)

(b)

(c)

FIG. 2: An example of time warping to determine local transformation between reference signal and testing signal, which are two consecutive sections drawn from a time series of ECG signals in slp60 record. (a) The original ECG $f(t)$ is plotted with the another signal $g(t')$before time warping. (b) The original ECG reference signal is plotted with testing signal after time warping. (c) The mapping between $t^{\star}$ and $t'^{\star}$ where the $\star$ stands for normalization, see discussion in section IV

II. PROBLEM FORMULATION: SHAPE SIMILARITY

In the application of classification, given two signals $F(t)$ and $G(t')$, where $t \in [0, T_1]$ and $t' \in [0, T_2]$. The signals were rescaled so that the maximum and minimum are 1 and 0:

$$f(t) = \frac{F(t) - \min(F(t))}{\max(F(t)) - \min(F(t))} \tag{1}$$

$$g(t) = \frac{F(t) - \min(G(t))}{\max(G(t)) - \min(G(t))} \tag{2}$$

The variables of t and t' are related by a function $\tau : [0, T_1] \to [0, T_2]$ where τ is some function given by

$$\tau = \arg\min_{\tau'(t)} \int_0^{T_1} dt \, |f(t) - g(\tau'(t))| \, . \tag{3}$$

In the analysis of real data, we consider a uniform sampling of the template signal $f(t)$ at $M + 2$ points at $t_i, i = 0, 1, \ldots, M + 1$. The testing signal $g(t')$ is also uniformly sampled at $N + 2$ points with equal spacing: $t_j, j = 0, 1, \ldots, N + 1$. The discrete version of $\tau(t)$ is then defined by

$$\{\tau_i\} = \arg\min_{\{\tau'_i\}} \sum_{i=0}^{M+1} |f_i - \hat{g}_i| \, , \tag{4}$$

where $f_i = f(t_i)$ and $\hat{g}\,(\tau'_i)$ is the linear interpolation of $\{g_j\}$ where $g_j = g(t_j)$. The time mappings τ_i are defined to be the set of mappings τ'_i that minimizes the absolute difference between sampled values of f_i and interpolated values of $\hat{g}_i$.

We illustrate the $\{\tau_i\}$ with an example. Consider $F(t) = 3\sin(2t)$ and $G(t') = \sin(t')$ in FIG.3, such that $t \in [0, \pi]$ and $t' \in [0, 2\pi]$. Suppose we choose $M = 3$ so that the interval between each sampling for f is $\Delta t = t_{i+1} - t_i = \pi/4$. We then have $M + 2 = 5$ samples are collected and normalized for $f : [f_0, f_1, f_2, f_3, f_4] = [0.5, 1, 0.5, 0, 0.5]$. Similarly, if we choose N=7, we have $N + 2 = 9$ samples for g_i. Note that T_1 and T_2 are π and 2π. By (4), we obtain $\{\tau_i\} = \{0, \pi/2, \pi, 3\pi/2, 2\pi\}$ for $i = 0, 1, 2, 3, 4$, thus recovering the time mapping to be a uniform scaling by a factor of 2. We will use this definition for time warping to analyze the pulses in ECG time series.

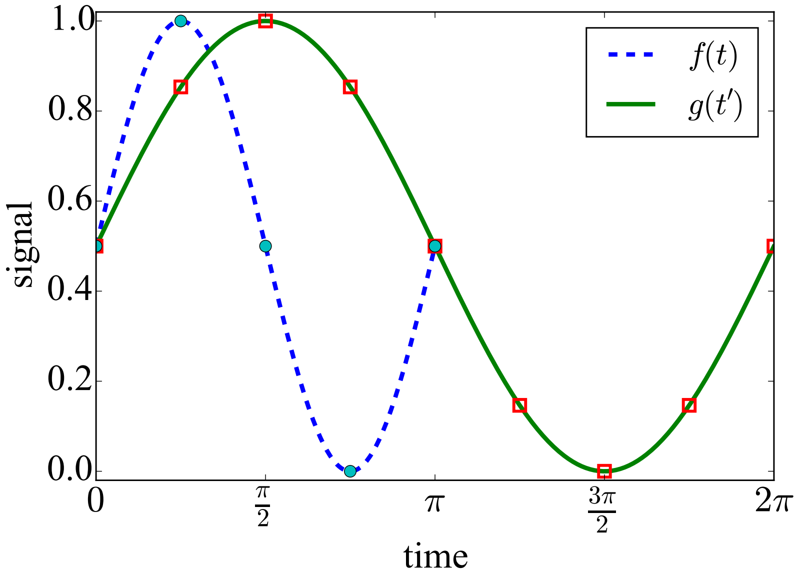


FIG. 3: An example illustrating the formulation of time mapping problem. The functions $f(t)$ and $g(t')$ are the normalized functions of $F(t) = 3\sin(2t)$ and $G(t') = \sin(t)$ respectively. The blue dots and red square markers represent the sampled values obtained at fixed time intervals of $f(t)$ and $g(t')$ respectively.

III. TIME WARPING GENETIC ALGORITHM

Genetic algorithm (GA) is a heuristic searching method, inspired by natural selection in evolutionary theory. It is quite useful for optimization problems that involve a large solution space. A population of N_p proposed solutions defined by different parameters is introduced, with each proposed solution defined as a chromosome with the parameters as encoded genes. These genes consist of loci and the phenotypes are expressed through the fitness function, which is the objective function of the optimization problem. Initially, a population of chromosomes is generated randomly and evolution of this population is driven mainly by elitism, which keeps the fittest individual chromosomes to the next generation. The change in the gene pool is generated through crossover and mutation. Crossover is used for exploration in the search for higher fitness and mutation is responsible for exploitation. In this section, we will introduce a type of GA that preserves the order of loci in the chromosome while maximizing the fitness function.

III.1. Solution Representation

Usually GA employs binary number to represent a solution. However, for optimization with real numbers, such as our time warping problem, the use of real numbers to represent solutions will be more efficient. In our work, every chromosome has M loci, and each locus is a real number τ_i. To initialize a chromosome, we generate M random floating points numbers between 0 and T_2 with uniform distribution. Then the numbers are reordered in ascending order. The initial population has N_p chromosomes. Using the previous example where $F(t) = 3\sin(2t)$ and $G(t) = \sin(t)$, every chromosome will have a length of 3: $[\lambda_1, \lambda_2, \lambda_3]$, where $T_2 = 2\pi \geq \lambda_3 \geq \lambda_2 \geq \lambda_1 \geq 0$.

III.2. Fitness Function

The fitness function of a chromosome l is defined to be the inverse of the summation of the absolute difference between $f(t_i)$ and $g(\tau_i)$:

$$F_l = \frac{1}{\sum_{i=1}^{M} |f(t_i) - \hat{g}_l(\tau_i)|^2} \tag{5}$$

Other metrics, such as absolute norm, can also be used. By maximizing F_l we obtain the time mappings between two functions f and g which satisfy shape similarity in (4). For our example involving the sine functions, the value of fitness for chromosome l can be calculated by $F_l = 1/(|f(t_1) - \hat{g}(\lambda_1)| + |f(t_2) - \hat{g}(\lambda_2)| + |f(t_3) - \hat{g}(\lambda_3)|) = 1/(|1 - \hat{g}(\lambda_1)| + |0.5 - \hat{g}(\lambda_2)| + |0 - \hat{g}(\lambda_3)|)$.

III.3. Parent Selection

Based on the idea of natural selection, chromosomes with higher fitness would have a higher probability to pass down their genetic material to the next generation. In our algorithm, every chromosome in the population is ranked so that $F_i \geq F_j$ for $i \leq j$. The parents are selected from the fittest $N_p - N_r$ chromosomes to produce N_r offspring chromosomes which replace chromosomes with low fitness. For the selection of each parent, the probability

P_l for chromosome l to be chosen is:

$$P_l = \frac{F_l}{\sum_{i=1}^{N_p-N_r} F_i} \tag{6}$$

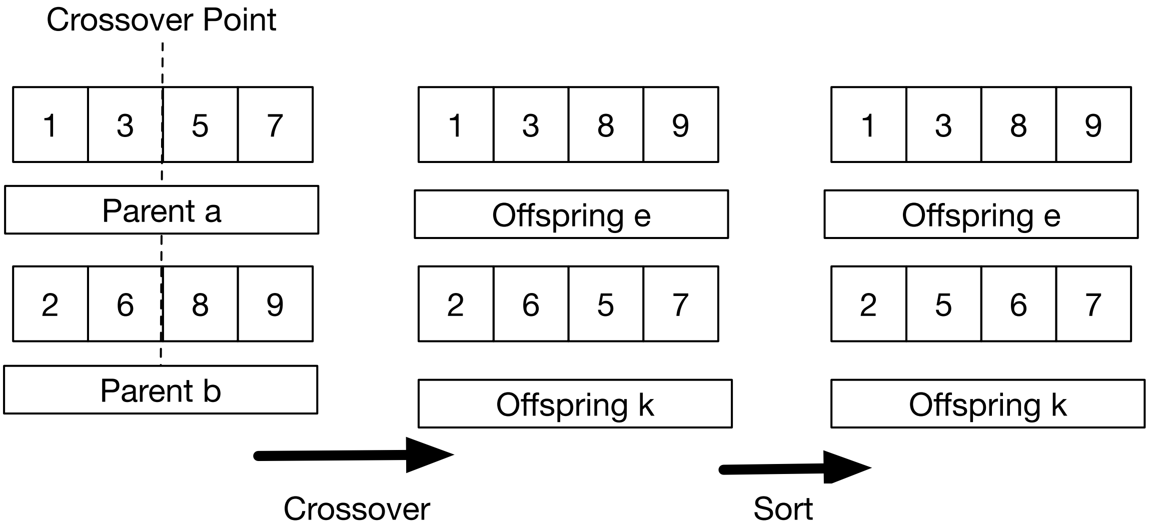


FIG. 4: An example of crossover operation at crossover point c = 3 involving single point crossover and sorting

III.4. Crossover Operation

We use single-point crossover operation in this work. Suppose the two parent chromosomes candidates are $[a_i]$ and $[b_i]$ and the two offspring chromosomes are denoted by $[e_i]$ and $[k_i]$, where $i = 1, 2, \ldots, M$. The crossover point $1 \leq c \leq M$ is randomly chosen with equal probability $1/M$. Then for the first offspring, we have $e_i = a_i$ for $i < c$ and $e_i = b_i$ for $i \geq c$; for the second offspring, $k_i = b_i$ for $i < c$ and $k_i = a_i$ for $i \geq c$. The loci in both offspring chromosomes will then be sorted and reordered in ascending order to make sure that $e_i \leq e_j$ and $k_i \leq k_j$ whenever $i < j$. An example is given in FIG.4. The offspring chromosomes produced then replace the weakest N_r chromosomes in the population.

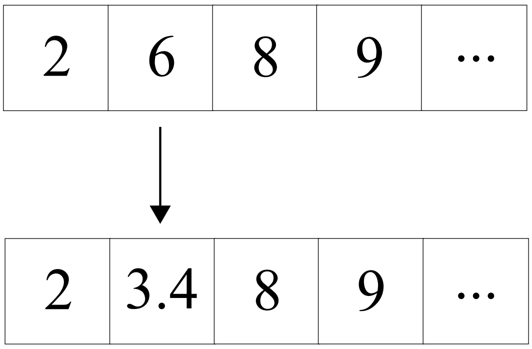


FIG. 5: An example of mutation operation at mutation point i = 2. For mutation, the changed value is bounded so that $l_3 = 8 \geq l_2 \geq l_1 = 2$.

III.5. Mutation Operation

In our algorithm, the chromosome with largest fitness value in the population will not be mutated. In chromosome l, the locus l_i has a probability P_m to mutate. The locus to be mutated will take a different value x between the values at l_{i-1} and l_{i+1} with uniform probability. If $i = 1$, the lower bound would be set to zero instead since that is the beginning position of a chromosome gene. Similarly, the upper bound of l_i is T_2 if $i = M$. An example is given in FIG.5.

III.6. Iteration

In each generation of the genetic algorithm, crossover operations and mutation operations will be performed on the population. The whole process is then iterated to search for the best solution. The iteration is terminated when a fixed number of generation is reached.

IV. APPLICATION

In this section, we will describe how to apply our proposed genetic algorithm on ECG signals of sleep apnea patients.

Again, the ECG data are obtained from MIT-BIH Polysomnographic Database[34] which can be accessed through `http://physionet.org`[35]. We mainly focus on the record `slp60` in the database. Here, the start and end of a quasi-period are determined by consecutive positions of R-peaks and positions of R-peaks are provided by the database. The sampling rate for ECG is 250Hz. The whole ECG time series is divided into 30-sec segments, and every segment has an associated sleeping stage (1, 2, 3, 4, REM or being awake), provided by the database. Also, every 30-sec segment is labelled as normal episode (NE), central apneic (CA) and obstructive apneic (OA) episodes, etc. We will focus on the NE, CA and OA episodes during sleep stage 1 of `slp60`. We denote the set of quasi-periods of ECG signals during NE (CA, or OA) episodes by S_{NE} (S_{CA}, S_{OA}), respectively.

IV.1. Data preprocessing and preparation

An ideal low-pass filter is applied with the cut-off frequency selected to be $(0.35)(250) = 87.5$Hz. In our analysis, $F(t)$ and $G(t')$ can be any signals from S_{NE}, S_{CA} or S_{OA}. Before applying time warping, the signals $F(t)$ and $G(t')$ are rescaled so that the maximum and minimum of the rescaled voltage signals $f(t)$ and $g(t')$ are 1 and 0.

IV.2. Time Warping

The time mapping $\tau(t_i)$ is obtained using the methodology described in section III. Linear regression is carried out for $\tau(t_i)$ against t_i to obtain the slope m and r^2 where r is the correlation coefficient. Both $\tau(t_i)$ and t_i are then normalized to be $\tau^\star(t_i)$ and $t_i^\star$ so that the values range from 0 to 1. The observable q^2, which is the linear deviation from straight line with unit slope, is defined as

$$q^2 = \frac{1}{M}\sum_{i=1}^{M} \left(\tau^\star(t_i) - t_i^\star\right)^2 , \tag{7}$$

where M is the length of chromosome. The value of M depends on the number of sampled points in signal $F(t)$. For a quasi-period in the ECG signal for `slp60`, M is around 200 since the sampling rate is 250Hz and we are mapping all the sampled points, except for the $F(t_0)$ and $F(T_1)$, to $G(t')$. We group the time mappings $\tau(t_i)$ in the following fashion: Suppose $F(t)$ is drawn from set S_X and $G(t')$ is drawn from set S_Y, where X and Y are NE, OA or CA, we denote the set of time mappings $\tau(t_i)$ between $F(t)$ and $G(t')$ to be (X-Y). Because both X and Y can be NE, OA or CA, we have nine sets of time mappings. Since sets of time mappings such as (NE-OA), (OA-NE) contain essentially the same information, we subtract three redundant sets of time mappings from the nine sets of time mappings to get six distinct sets of time mappings, consisting of the "intra-set" with three members: (NE-NE), (OA-OA) and (CA-CA) and the "inter-set" with three members: (NE-OA), (NE-CA) and (OA-CA).

As for the parameters in our TWGA, we set the population size in each generation to be $N_p = 100$ and the number of chromosomes replaced in each generation to be $N_r = 50$. The mutation probability for each locus is $P_m = 0.02$ and the iteration is terminated at 1000th generation for the search of each time mapping. For each set of time mappings (X-Y), we randomly select one sample $F(t)$ from S_X and one sample $G(t')$ from S_Y and obtain the time mapping between $F(t)$ and $G(t')$ using TWGA. We then repeat this process for 1000 times to obtain a distribution of time mappings in (X-Y). Therefore, a distribution of r^2 and q^2 is collected for each set of time mappings (X-Y), such that r^2 is the coefficient of determination from linear regression and q^2 is the linear deviation as defined in (7).

IV.3. Statistical Analysis

We would like to investigate if there exist differences with statistical significance in the distribution of observables obtained from different sets of time mappings. Kruskal-Wallis H test [36] is a non-parametric statistical test used to determine if the population median of all groups are the same. The null hypothesis for Kruskal-Wallis Test, H_0, states that the samples of different groups are drawn from the same population by uniform sampling, whereas the alternative hypothesis, H_1, states that at least one of the groups have a different population median. Therefore, we can use the results of Kruskal-Wallis H test to reject H_0. If the p-value obtained from Kruskal-Wallis H test is less than 0.05, we should reject H_0 and accept the alternative hypothesis H_1. If Kruskal-Wallis H test supports H_1, Mann-Whitney U test[37] is applied for pairwise post hoc testing to determine which of the groups are different from the rest of the population, as observed by Kruskal-Wallis H test. Since type I error, which is the false rejection of a null hypothesis, increases under this algorithm for multiple comparison, the critical p-value for Mann-Whitney U test is modified by Bonferroni correction [38], which states that α should be divided by C_2^h, which is the number of possible pairs among the groups, if we are testing for h groups.

We denote the distribution of linear deviation q^2 from the time mappings in set (X-Y) by $P_{X-Y}(q^2)$. Similarly, the distribution of coefficient of determination r^2 from the time

TABLE I: Results of Kruskal-Wallis H Test recording the H-value, F-value and the corresponding confidence P-value

	H-value	P-value
q^2	25.64	1.046e-04
r^2	21.89	5.476e-04

mappings in set (X-Y) is denoted by $P_{X-Y}(r^2)$. Kruskal-Wallis H Test is applied on both types of distribution.

V. RESULTS

Signals $F(t)$ and $G(t')$ were drawn from the three sets of signals that occurs during sleep stage 1: normal (S_{NE}), obstructive apnea (S_{OA}) and central apnea (S_{CA}). We are interested in obtaining the distributions of two variables: $P_{X-Y}(r^2)$ and $P_{X-Y}(q^2)$. The coefficient of determination r^2 from linear interpolation of the time mapping function is the simply the square of correlation coefficient. The symbol q^2 denotes linear deviation, define in (7) as deviation of the $\tau_i^\star$ value from $t_i^\star$.

Kruskal-Wallis test is applied on the distributions $P_{X-Y}(q^2)$. The results are tabulated in Table I which shows that Kruskal-Wallis H Test supports the alternative hypothesis H_1 for the distribution of linear deviation. It follows that for $P_{X-Y}(q^2)$ there is at least one group with a population median different from the rest of the groups with statistical significance at confidence level $\alpha = 0.05$ as the P-value is less than α. Alternatively, since the value of the test statistics, or the H-value, is larger that the critical value $H^c = 11.07$ corresponding to $\alpha = 0.05$, the null hypothesis is rejected. Similarly, Kruskal-Wallis H Test is also applied on the distributions $P_{X-Y}(r^2)$, and the test supports the alternative hypothesis H_1 that for $P_{X-Y}(r^2)$ there is at least one group with a population median different from the rest of the groups with statistical significance at confidence level $\alpha = 0.05$.

We therefore proceed to carry out post hoc test for both linear deviation q^2 and coefficient of determination r^2. Because we have 6 sets of time mappings corresponding to 6 explanatory variables, there are altogether fifteen cases for pairwise testing considering that the pair (X$_1$-

Y_1) and (X_2-Y_2) is the same as the pair (X_2-Y_2) and (X_1-Y_1). For each case, we compare (X-Y) with (X'-Y').

For $P_{X-Y}(q^2)$, since Kruskal-Wallis H test rejects H_0, we carry out Mann-Whitney U Test for further analysis following Kruskal-Wallis H test to find out which of the groups is responsible for the deviation from the entire population. The results are shown in Table II. The P-value has to be less than the adjusted value of $\alpha = 0.0033$ by Bonferroni correction for statistical significant deviations. From the table, case 1: (NE-NE) with (OA-OA), case 6:(OA-OA) with (CA-CA), case 7:(OA-OA) with (NE-OA), case 8:(OA-OA) with (NE-CA) and case 9:(OA-OA) with (CA-OA) show statistically significant differences in sample medians. This indicates the phase shift between intra-set time mappings of signals extracted from OA episodes is different not only from that obtained in other intra-set time mappings but also from that obtained in all inter-set time mappings. Intra-set time mappings (X-X) including (NE-NE), (OA-OA) and (CA-CA) refers to comparing two signals drawn from the same set S_X; whereas inter-set time mappings (X-Y) such as (NE-OA), (NE-CA) and (OA-CA) refers to comparing a signal in set S_X with another signal in a different set S_Y.

On the other hand, since Kruskal-Wallis H test also rejects H_0 for $P_{X-Y}(r^2)$, Mann-Whitney U Test is again carried out as a post hoc test. The results are shown in Table III. Contrast to the analysis of q^2, case 1 shows no statistical dominance over the whole population. The distribution of r^2 during OA episodes shows no statistically significant deviation from that obtained during normal episodes. Nonetheless, all other cases involving (OA-OA) still shows deviations with statistical significance. Therefore, observable r^2 also captures the difference in phase distortion between signals obtained during OA periods and signals obtained for inter-set mappings.

VI. DISCUSSION

In our short study, we have demonstrated the use of TWGA as a tool to study the phase difference. We then apply the algorithm on the `slpdb60` ECG signals, and show that ECG signals obtained during different sleeping states exhibit different phase shifts as captured by linear deviation q^2 and coefficient of determination r^2 from linear interpolation. In particular, ECG signals obtained during OA episodes show statistically significant differences from signals obtained during CA episodes and during NE periods in terms of r^2 and q^2

TABLE II: Results of Mann-Whitney U test on as post hoc test for Kruskal-Wallis H test on coefficient of determination q^2

Case	(X-Y)	(X′-Y′)	U-value	P-value
1	(NE-NE)	(OA-OA)	464651.0	6.19e-03
2	(NE-NE)	(CA-CA)	488780.0	3.85e-01
3	(NE-NE)	(NE-OA)	482836.0	1.84e-01
4	(NE-NE)	(NE-CA)	482412.0	1.73e-01
5	(NE-NE)	(CA-OA)	483591.0	2.04e-01
6	(OA-OA)	(CA-CA)	453132.0	2.84e-04
7	(OA-OA)	(NE-OA)	447235.0	4.39e-05
8	(OA-OA)	(NE-CA)	446820.0	3.82e-05
9	(OA-OA)	(CA-OA)	448262.0	6.16e-05
10	(CA-CA)	(NE-OA)	494213.0	6.54e-01
11	(CA-CA)	(NE-CA)	494229.0	6.55e-01
12	(CA-CA)	(CA-OA)	495152.0	7.07e-01
13	(NE-OA)	(CA-OA)	499643.0	9.78e-01
14	(NE-OA)	(NE-CA)	499746.0	9.84e-01
15	(NE-CA)	(CA-OA)	499473.0	9.67e-01

related to phase distortion extracted by TWGA. This may facilitate the development of autonomous classification systems incorporating extra information regarding phase shifts that differentiates obstructive apneic ECG signals from both central apneic signals and normal signals with enhanced accuracies. Such systems may greatly reduce medical resources allocated for diagnosis of sleep apnea among other diseases. Further statistical study is required on an expanded database with more patients with different conditions for the

TABLE III: Results of Mann-Whitney U test on as post hoc test for Kruskal-Wallis H test on coefficient of determination r^2

Case	(X-Y)	(X′-Y′)	U-value	P-value
1	(NE-NE)	(OA-OA)	483246.0	1.94e-01
2	(NE-NE)	(CA-CA)	475194.0	5.47e-02
3	(NE-NE)	(NE-OA)	469188.0	1.70e-02
4	(NE-NE)	(NE-CA)	475277.0	5.56e-02
5	(NE-NE)	(CA-OA)	473288.0	3.86e-02
6	(OA-OA)	(CA-CA)	457914.0	1.12e-03
7	(OA-OA)	(NE-OA)	452592.0	2.41e-04
8	(OA-OA)	(NE-CA)	457729.0	1.06e-03
9	(OA-OA)	(CA-OA)	456755.0	8.11e-04
10	(CA-CA)	(NE-OA)	493437.0	6.11e-01
11	(CA-CA)	(NE-CA)	499610.0	9.76e-01
12	(CA-CA)	(CA-OA)	497250.0	8.31e-01
13	(NE-OA)	(CA-OA)	494082.0	6.47e-01
14	(NE-OA)	(NE-CA)	495749.0	7.42e-01
15	(NE-CA)	(CA-OA)	498523.0	9.09e-01

purpose of classifying different apneic episodes and understanding the biological process in sleep apnea. Characteristics of quasi-periodic biological time-series that are universal among different patients can be extracted and these common signal features may provide insights in the mechanisms of sleep apnea and other diseases such as arrhythmia. Moreover, other measures of phase changes may be considered apart from q^2 and r^2 in order to better understand the instantaneous local changes in a quasi-period. This allows TWGA to be utilized as a tool to observe the variation of phase shift as a dynamic process. Furthermore, the information of local phase shifts obtained by TWGA can be useful in embedding of time-series in the phase space. A limitation of this algorithm involves the fluctuation of signal strength as mentioned in section V. A possible remedy would be to use some non-constant scaling functions to normalize signal F and G instead of using a fixed scale factor. Another approach is to study different components of one quasi-period (such as either p-wave, QRS complex or T-wave in an ECG signal) seperately[33]. Alternatively, we can use the difference of derivatives of the testing signal and reference signal instead of the difference in magnitude[39, 40]. Since this requires the first derivative of the functions to be continuous, linear interpolation described in section III should be replaced by cubic spline interpolation.

VII. CONCLUSION

In this paper, we have developed a novel algorithm for shape matching across different signals in the time domain with genetic algorithm(TWGA). TWGA can be used to more accurately study the phase shift in a quasiperiodic function. In particular, we apply the algorithm on the ECG signals of a sleep apnea patient and collect statistical data for analysis. We show that the distribution of phase-related variables obtained during obstructive apneic episodes (OA) is different from that obtained during central apneic (CA) and normal episodes (NE).

ACKNOWLEDGMENT

K. Y. Szeto acknowledges the support of grant FSGRF13SC25 and FSGRF14SC28.

[1] C. Alexakis, HO Nyongesa, R. Saatchi, ND Harris, C. Davies, C. Emery, RH Ireland, and SR Heller. Feature extraction and classification of electrocardiogram (ECG) signals related to hypoglycaemia. pages 537–540. IEEE, 2003. Conference Proceedings.

[2] Pedro Fonseca, Xi Long, Mustafa Radha, Reinder Haakma, Ronald M. Aarts, and JÃĺrÃťme Rolink. Sleep stage classification with ECG and respiratory effort. *Physiological Measurement*, 36(10):2027, 2015.

[3] Ernesto Pereda, Rodrigo Quian Quiroga, and Joydeep Bhattacharya. Nonlinear multivariate analysis of neurophysiological signals. *Progress in Neurobiology*, 77(1âĂŞ2):1–37, September 2005.

[4] Yasushi Makihara, Ngo Thanh Trung, Hajime Nagahara, Ryusuke Sagawa, Yasuhiro Mukaigawa, and Yasushi Yagi. Phase Registration of a Single Quasi-Periodic Signal Using Self Dynamic Time Warping. In Ron Kimmel, Reinhard Klette, and Akihiro Sugimoto, editors, *Computer Vision âĂŞ ACCV 2010*, number 6494 in Lecture Notes in Computer Science, pages 667–678. Springer Berlin Heidelberg, November 2010. DOI: 10.1007/978-3-642-19318-7_52.

[5] Roger C. Stone, Graeme L. Hammer, and Torben Marcussen. Prediction of global rainfall probabilities using phases of the Southern Oscillation Index. *Nature*, 384(6606):252–255, November 1996.

[6] Prodromos Tsinaslanidis, Antonis Alexandridis, Achilleas Zapranis, and Efstratios Livanis. Dynamic Time Warping as a Similarity Measure: Applications in Finance. (Journal Article), 2014.

[7] Yevgeniy Bodyanskiy and Sergiy Popov. Neural network approach to forecasting of quasiperiodic financial time series. *European Journal of Operational Research*, 175(3):1357–1366, December 2006.

[8] George B. Gelfreikh, Yury A. Nagovitsyn, and Elena Yu Nagovitsyna. Quasi-Periodic Oscillations of Microwave Emission in Solar Active Regions. *Publications of the Astronomical Society of Japan*, 58(1):29–35, February 2006.

[9] Norden E. Huang, Zheng Shen, Steven R. Long, Manli C. Wu, Hsing H. Shih, Quanan Zheng, Nai-Chyuan Yen, Chi Chao Tung, and Henry H. Liu. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 454, pages 903–995. The Royal Society, 1998.

[10] Donald J. Berndt and James Clifford. Using Dynamic Time Warping to Find Patterns in Time Series. volume 10, pages 359–370. Seattle, WA, 1994. Conference Proceedings.

[11] Lin Feng, Xiaoyan Zhao, Yiwei Liu, Yuan Yao, and Bo Jin. A similarity measure of jumping dynamic time warping. volume 4, pages 1677–1681. IEEE, 2010. Conference Proceedings.

[12] Tanveer Syeda-Mahmood, David Beymer, and Fei Wang. Shape-based matching of ECG recordings. In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual*

International Conference of the IEEE, pages 2012–2018. IEEE, 2007.

[13] Chotirat Ann Ratanamahatana and Eamonn Keogh. Making time-series classification more accurate using learned constraints. SIAM, 2004. Conference Proceedings.

[14] Sergei L. Kosakovsky Pond, David Posada, Michael B. Gravenor, Christopher H. Woelk, and Simon D. W. Frost. GARD: a genetic algorithm for recombination detection. *Bioinformatics*, 22(24):3096–3098, December 2006.

[15] Yong-Sheng Ding and Tong-Liang Zhang. Using ChouâĂŹs pseudo amino acid composition to predict subcellular localization of apoptosis proteins: An approach with immune genetic algorithm-based ensemble classifier. *Pattern Recognition Letters*, 29(13):1887–1892, October 2008.

[16] Peter Clote. An efficient algorithm to compute the landscape of locally optimal RNA secondary structures with respect to the Nussinov-Jacobson energy model. *Journal of computational biology*, 12(1):83–101, 2005.

[17] David J. Wales, Jonathan PK Doye, Mark A. Miller, Paul N. Mortenson, and Tiffany R. Walsh. Energy landscapes: from clusters to biomolecules. *Advances in Chemical Physics*, 115:1–112, 2000.

[18] David J. Wales, Mark A. Miller, and Tiffany R. Walsh. Archetypal energy landscapes. *Nature*, 394(6695):758–760, 1998.

[19] Jonathan PK Doye. Network topology of a potential energy landscape: A static scale-free network. *Physical review letters*, 88(23):238701, 2002.

[20] Pablo G. Debenedetti and Frank H. Stillinger. Supercooled liquids and the glass transition. *Nature*, 410(6825):259–267, 2001.

[21] Anil N. Patel, David Davis, Charles Foster Guthrie, David Tuk, Tai Thien Nguyen, John Williams, and others. Optimizing Cyclic Steam Oil Production with Genetic Algorithms. In *SPE Western Regional Meeting*. Society of Petroleum Engineers, 2005.

[22] Changliang Xia, Peijian Guo, Tingna Shi, and Mingchao Wang. Speed control of brushless DC motor using genetic algorithm based fuzzy controller. In *Proceeding of the 2004 International Conference on Intelligent Mechatronics and Automation, Chengdu, China, 3rd edn. A Treatise on Electricity and Magnetism*, volume 2, pages 68–73, 2004.

[23] King Loong Shiu and Kwok Yip Szeto. Self-adaptive Mutation Only Genetic Algorithm: An Application on the Optimization of Airport Capacity Utilization. In *Intelligent Data*

Engineering and Automated LearningâĂŞIDEAL 2008, pages 428–435. Springer, 2008.

[24] Chris Messom. Genetic algorithms for auto-tuning mobile robot motion control. 2002.

[25] Kwok Yip Szeto and L. Y. Fong. How adaptive agents in stock market perform in the presence of random news: A genetic algorithm approach. In *Intelligent Data Engineering and Automated LearningâĂŤIDEAL 2000. Data Mining, Financial Engineering, and Intelligent Agents*, pages 505–510. Springer, 2000.

[26] L. Y. Fong and K. Y. Szeto. Rules extraction in short memory time series using genetic algorithms. *The European Physical Journal B-Condensed Matter and Complex Systems*, 20(4):569–572, 2001.

[27] Mohamed Benbouziane. Portfolio selection using genetic algorithm. (Journal Article), 2012.

[28] R. Jiang, K. Y. Szeto, Y. P. Luo, and D. C. Hu. Distributed parallel genetic algorithm with path splitting scheme for the large traveling salesman problems. In *Proceedings of Conference on Intelligent Information Processing, 16th World Computer Congress*, pages 21–25, 2000.

[29] Pankaj Kumar, Ankur Gupta, Valadi K. Jayaraman, and BhaskarD Kulkarni. Aligning time series with genetically tuned dynamic time warping algorithm. In *Advances in Metaheuristics for Hard Optimization*, pages 251–261. Springer, 2008.

[30] Volkan Tuzcu and Selman Nas. Dynamic time warping as a novel tool in pattern recognition of ECG changes in heart rhythm disturbances. volume 1, pages 182–186. IEEE, 2005. Conference Proceedings.

[31] BS Raghavendra, Deep Bera, Ajit S. Bopardikar, and Rangavittal Narayanan. Cardiac arrhythmia detection using dynamic time warping of ECG beats in e-healthcare systems. pages 1–6. IEEE, 2011. Conference Proceedings.

[32] MJ Burke and GP Shorten. Characterisation of ECG profile timing using dynamic time warping. *Recent Advances in Telecommunications and Circuit Design*, (Journal Article):29–34, 2013.

[33] Sofiane Boudaoud, Conor Heneghan, HervÃľ Rix, Olivier Meste, and Ciara O'Brien. P-wave shape changes observed in the surface electrocardiogram of subjects with obstructive sleep apnoea. In *Computers in Cardiology, 2005*, pages 359–362. IEEE, 2005.

[34] Y. Ichimaru and G. B. Moody. Development of the polysomnographic database on CD-ROM. *Psychiatry and Clinical Neurosciences*, 53(2):175–177, 1999.

[35] Ary L. Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M. Hausdorff, Plamen Ch Ivanov, Roger G. Mark, Joseph E. Mietus, George B. Moody, Chung-Kang Peng, and H. Eugene Stanley. Physiobank, physiotoolkit, and physionet components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000.

[36] William H. Kruskal and W. Allen Wallis. Use of ranks in one-criterion variance analysis. *Journal of the American statistical Association*, 47(260):583–621, 1952.

[37] Henry B. Mann and Donald R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, pages 50–60, 1947.

[38] Eric S. Lander and David Botstein. Mapping mendelian factors underlying quantitative traits using RFLP linkage maps. *Genetics*, 121(1):185–199, 1989.

[39] Yang Zhang and Thomas F. Edgar. A robust dynamic time warping algorithm for batch trajectory synchronization. pages 2864–2869. IEEE, 2008. Conference Proceedings.

[40] Eamonn J. Keogh and Michael J. Pazzani. Derivative Dynamic Time Warping. In *SDM*, volume 1, pages 5–7. SIAM, 2001.

Topological Evolution of Financial Network: A Genetic Algorithmic Approach

Ga Ching Lui, Chun Yin Yip, Kwok Yip Szeto

Abstract

The structure of financial market is captured using a novel time warping method known as discrete time warping genetic algorithm (dTWGA). In contrast to previous studies which estimate the correlations between different time series, dTWGA can be used to analyse time series with different lengths and with data sampled unevenly. Moreover, since coupling between different time series or at different periods of time would be changing over time, the time delay for the influence of a time series to reach another time series would be changing as well, which would not be well captured with correlation measurements. The proposed algorithm is applied on Dow Jones Index (DJI) and its compositions consisting of 30 stocks, and different measurements are performed to observe the evolution of the network structure. It is suggested that there are major topological changes during market crashes, leading to a significant decrease in the size of the network.

I. INTRODUCTION

There have been rapid developments in network science and graph theory in recent years as the field is gaining more attention, with numerous applications in different sectors ranging from ecology [10], epidermic spreading [3], interbank network [2], to information propagation through social media [17]. In finance, pairwise correlation of financial time series has been used to build a description of the financial network by treating each stock as a node in the network and by using some measurements of the coupling strength as the weight of the edges [4, 5, 15]. In particular, the hierarchical structure of a financial network is constructed using correlation as a similarity measurement to describe the coupling between different stocks, with close proximity between two stocks representing a positive correlation, while stocks that are very far from the others are negatively correlated [9]. Onnela [12] further applies this methodology to study changes of S&P500 and its components in relation to Black Monday, a financial crisis in 1987. However, there exists limitations with this methodology. For instance, Pearson correlation cannot be used to describe non-linear relations, and thus the network structure described with the above method would not be able to detect interactions of this type. Furthermore, there may be time lags between the occurrence of a price change in a stock and the information of such a change reaching another node to incur certain effects on the price of that node, and would therefore evade the detections made with Pearson correlation. While measuring the cross-correlation with time delay may help determine the time delay and thus rediscover this type of causal relation, the time delay itself may not actually be time-independent and may be affected by the market environment or the investment climate at that time, and therefore non-linear mapping between the two time series is required. In extreme cases, if the time delay becomes negative, then causality between the two stocks is reversed. The third limitation has to do with the length of the time series and the way the data are gathered. For high-frequency trading, the data points are not evenly sampled as transactions are made at different instances [4]. In addition, time series collected for different stocks might not have the same length. Thus, preprocessing of the data points, such as addition or removal of data points, prior to calculating the correlations would be required, and this in turn leads to more assumptions and bias made about the system. In this paper, we are motivated to rectify some of these problems with a new methodology involving genetic algorithm.

Genetic algorithm (GA) is meta-heuristic searching method employed in optimization problems with large searching space. GA has been utilized across different disciplines such as game theory [18], portfolio selection [13], biological time series [7], astrophysics [11] and network optimization [6, 8]. In this work, we use GA as a means to recover the coupling between different stocks and this method is called discrete time warping genetic algorithm (dTWGA), which is the discrete version of the method TWGA we have previously proposed [7]. Different from TWGA, dTWGA does not involve the crossover operator. GAs without crossovers have previously been studied, with the example of Mutation Only Genetic Algorithm (MOGA) which involves the use of mutation matrix and can be seen as a generalization of traditional GAs [14]. Making use of dTWGA, the coupling with non-linear time delay can be described and is used to construct a description of the financial market. Topological changes of such network across time is compared with the performance of financial market as a whole. In particular, the network size shrinks significantly at the time of crashes. This paper is organized as follows: The ideas and methodology of dTWGA are explained in Sect. II. For Sect. III, the procedure of recovering the network structure is described, and the application is documented in Sect. IV. The results are presented and discussed in Sect. V, and the paper concludes with Sect. VI.

II. DISCRETE TIME WARPING GENETIC ALGORITHM (DTWGA)

Suppose we are given two time series, namely X(t) and Y(t), and the objective of the proposed method dTWGA is to obtain the time mappings $\tau(t)$ such that the overlapping area of the graphs in the time domain for the $X(\tau(t))$ and $Y(t)$ is maximized. Such non-linear transformation is referred to here as time warping. Discrete TWGA aims to tackle this problem by partitioning the discrete time domain with as few constraints as possible since the determination of constraints requires prior knowledge or domain knowledge of the systems concerned, and we would like to minimize the number of assumptions made. Due to the large solution space, genetic algorithm is used to minimize the sum of absolute pairwise distance of $X(\tau(t_j) = t_i)$ and $Y(t_i)$ across all t_i.

II.1. Solution Representation

Defining the two normalized time series to be $S_i = \{S_i(n)|n = 0, 1, ..., N, N + 1\}$ and $S_j = \{S_j(m)|m = 0, 1, ..., M, M + 1\}$, where n and m are the time indices, while $\max(S_i) = \max(S_j) = 1$ and $\min(S_i) = \min(S_j) = 0$, a solution in the searching space is encoded in dTWGA to be chromosome k which consists of N time indices such that $C_k = \{C_k[l]||l = 1, 2, ..., N\}$. Each chromosome is of length N instead of $N + 1$ because the boundary constraint is imposed such that $S_i(0)$ is always mapped to $S_j(0)$ while $S_i(N + 1)$ is always mapped to $S_j(M + 1)$. An example of a chromosome is shown in Fig. 1. Two constraints are considered when a population of chromosomes evolves:

Monotonicity: $C_k[l] \leq C_k[l + 1] \quad \forall_l$

Boundary: $0 \leq C_k[l] \leq M + 1 \quad \forall_l$

To initialize a population of chromosomes, N integers are drawn randomly from a uniform deviate in the closed set $[0, M + 1]$ for each chromosome. The N integers are then sorted in ascending order to form a chromosome. N_p chromosomes are produced with this method to form a population such that $k = 1, 2, ..., N_p$ for chromosome C_k.

II.2. Mutation

For each generation, N_m loci in each chromosome are selected to mutate, except for the fittest chromosome. This is known as elitism, which ensures that the best solution or the fittest chromosome in the population remains in the next generation. The concept of fitness will be clarified in the next section. For a selected position l_m, the original value $C_k(l_m)$ is changed to a random integer drawn from a uniform deviate in $[C_k(l_m - 1), C_k(l_m + 1)]$ unless $l_m = 1$ or N. If $l_m = 1$, an integer is selected from $[0, C_k(2)]$. If $l_m = N$, an integer is selected from $[C_k(N - 1), M + 1]$. This ensures that both monotonicity and boundary constraints are satisfied. A total of $N_m \times (N_p - 1)$ mutates in each generation and this corresponds to the mutation probability of N_m/N. An example of mutation is shown in Fig. 2.

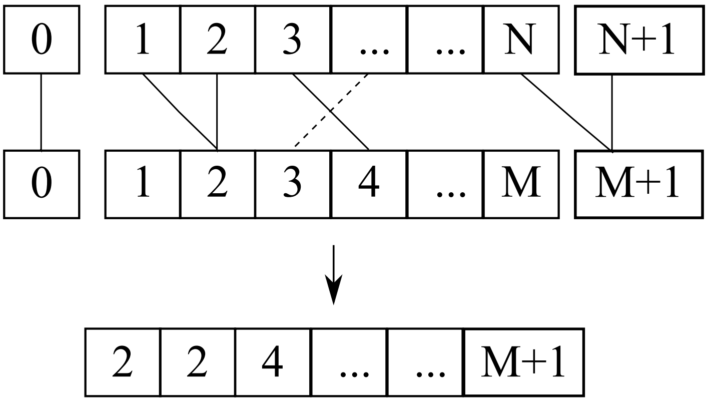


FIG. 1: An example of chromosome of length N representing the mapping of the time indices n and m. The mappings for the two time series are represented by solid line, whereas the dotted line denotes the mapping that is not allowed by the monotonicity requirement. The boundaries are fixed and are therefore not included in the chromosome sequence. This mapping is encoded in the chromosome as shown in the bottom row of numbers in the figure.

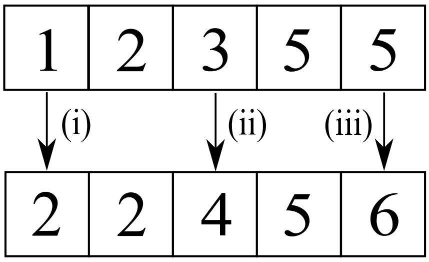


FIG. 2: An example of mutation with $N_m = 3$ and $N = 5$, corresponding to a mutation probability of 0.6. For mutation (i), (ii) and (iii), the modified values are chosen from the closed sets [0,2], [2,5] and [5,6] respectively.

II.3. Fitness

To define concept of fitness, we consider the cost function of chromosome k, denoted as s_k. The fitter chromosomes would correspond to lower values of cost, while the weaker chromosomes would have higher values of cost. Since the objective of this algorithm is to minimize the distance between two time series through time warping, we define the cost function to be

$$s_k = |S_i(0) - S_j(0)| + |S_i(N+1) - S_j(M+1)| + \sum_{l=1}^{N} |S_i(l) - S_j(C_k[l])| \ . \qquad (1)$$

Although we simply use absolute norm here, other norms can also be used. By allowing the population to evolve, the cost of the best chromosome would decrease using suitable selection schemes.

II.4. Selection

N_p chromosomes in the population are sorted in ascending order according to the cost function $s_k \leq s_{k+1} \forall_k$. This means that the fitter chromosomes would be assigned with smaller values of k. N_k chromosomes that has the highest cost would be replaced by fitter chromosomes with lower costs. To achieve this, chromosome C_k is removed for $k \geq N_p - N_k$. N_k selections are made in order to replace the removed chromosomes. For each selection, a chromosome is chosen from the fitter chromosomes with $k \leq N_p - N_k$ according to the following probability:

$$P(C_k) = \frac{\sum\limits_{\substack{i=1 \\ i \neq k}}^{N_p - N_k} s_i}{(N_p - N_k - 1) \sum\limits_{j=1}^{N_p - N_k} s_j} \ . \qquad (2)$$

The probabilities defined in (2) are normalized by default. According to the equation, chromosomes with lower costs would have a higher probability to be chosen. The chosen chromosome would replace one of the removed chromosomes, thus lowering the average cost of the population.

II.5. Iteration

The processes of mutation, fitness evaluation, sorting and selection are iterated to produce new generations of chromosomes and to allow the population to evolve until a stopping criteria is reached. In our algorithm, the stopping criteria is defined as the fixed number of generations produced. When the G-th generation is reached, the best chromosome in that population is returned to be the time mapping between the two time series, and the distance $d(S_i, S_j)$ between S_i and S_j is returned to be $s_{k=1}$ in the G-th generation. Since $M = N$

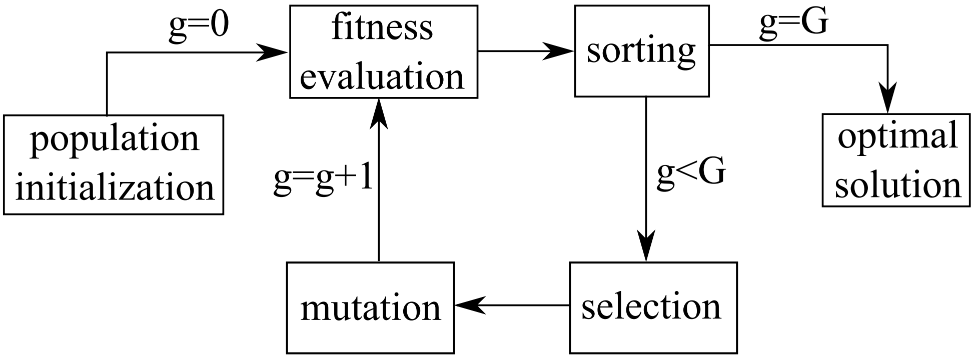


FIG. 3: Flow chart of dTWGA, consisting of operators that maintain monotonicity, with the number of generations reached to be the stopping criteria, and that g denote the number of generations.

and both are fixed in this work, normalization of the distance is not required. If $M \neq N$, then the distance should be normalized by a constant factor of $N+2$. A flow chart of this algorithm is shown in Fig. 3.

III. FINANCIAL NETWORK CONSTRUCTION

Given a stock i, consider the time series of stock price P_i, we can detrend the data using log return S_i, which is typically defined as

$$S_i(t) = \ln P_i(t) - \ln P_i(t-1) \ . \tag{3}$$

The network structure of N_s stocks can be captured using dTWGA. The nodes in the graph are the stocks, while the edge weightings between stock i and stock j are given by

$$d_{ij} = \min\left(d(S_i, S_j), d(S_i, -S_j)\right)) \ , \tag{4}$$

where $d(S_i, S_j)$ is a measure of positive correlation, while $d(S_i, -S_j)$ is a measure of negative correlation. This measurement is similar to that suggested by Tsinaslanidis [16] for dynamic time warping. By this construction, $d_{ij} = 0$ if $i = j$ and $\forall_{i,j}, d_{ij} \geq 0$. As d_{ij} should be the same as d_{ji}, we only need to evaluate $N_s(N_s-1)/2$ entries in the $N_p \times N_p$ adjacency matrix $\mathbf{D}$, while each entry requires evaluation of two distances using dTWGA, resulting in a weighted, undirected and connected network, of which the topology can be captured using the three measurements described below.

III.1. Minimum Spanning Tree

A minimum spanning tree (MST) refers to an acyclic subgraph with N_p nodes and $N_p - 1$ edges that has the smallest total weight summed over all edges compared to the total weight of other spanning trees. In other words, we may describe the size of an MST, D_{min}, with the tree length:

$$L = \sum_{(i,j) \in D_{min}} d_{ij} \ , \tag{5}$$

where (i, j) is the edge connecting node i and node j. An MST can be constructed using Kruskal's algorithm which is a type of greedy algorithm:

Pseudo-code of Kruskal's algorithm

```
D_min = empty graph
E = list of all edges
sort E in ascending order
for e in E:
   if (D_min + e) is acyclic:
      add e to D_min
```

We would measure tree length L to characterize a network at different time periods in the later sections.

III.2. Maximum Degree Ratio

In the MST, we are interested in characterizing the largest hub, which has the largest number of nearest neighbours. At crashes, it is expected that the hubs would be more influential to other nodes, and therefore it would be interested to observe the weighted degree of network hubs in MST. We define the maximum degree ratio as

$$\xi(D_{min}) = \frac{\text{weighted degree of node } i}{\text{number of nearest neighbours of node } i} \ , \tag{6}$$

where node i is the node with the largest number of neighbours in the MST, and the weighted degree is simply the sum of weights of all edges connected to node i in MST.

III.3. Spectrum

We would compare the spectrum at different time periods. Given an adjacency matrix, we denote the eigenvalues to be λ_i, where $i = 1, 2, ..., N_s$, such that $\forall_i$, $\lambda_i \geq \lambda_{i+1}$. In this work, we would only compare the largest eigenvalue λ_1 of matrix $\mathbf{D}$ with the market environment.

IV. EXPERIMENT

The algorithm is applied to analyse the daily closing price of Dow Jones Index (DJI) and its components consisting of 30 stocks during the period from mid-March in 2008 to the beginning of October in 2016, and the data are obtained from Yahoo [1]. A histogram of log return obtained during this period is plotted in Fig. 4, showing a bell-shaped distribution centred around $S_{dji} = 0$, which is one of the nice properties of using log return, with outliers representing large crashes and sudden increase of stock prices in the financial market. The largest price fluctuation during this period occurs between 2008 and 2009, during which the global financial crisis was triggered. There are also some moderate price fluctuations during 2010, 2011, and 2015 and 2016. To monitor the changes of the financial network, we need to infer the coupling between different stocks. The distances obtained with dTWGA are estimations of the couplings between the components, giving a 30×30 adjacency matrix $\mathbf{D}$ with entries d_{ij}.

A moving window with size w is applied on the time series to obtain the network structure at different time frames. For dTWGA, we are therefore evaluating the distance d_{ij} between $\{S_i(n')|n' = t, t+1, ..., t+w\}$ and $\{S_j(m')|m' = t, t+1, ..., t+w\}$ by (4). By shifting the time axis such that $n = n' - t$ and $m = m' - t$, we can then follow the algorithm described in Sect. II and III accordingly, with $M = N = w - 2$. For the parameters in this work, the size of population N_p is set to be 8 chromosomes, the number of removed chromosomes in each generation is set to be $N_k = 4$, and the mutation parameter is set to be $N_m = 3$. Here, the window size is set to be $w = 20$, which is around 1 month. This corresponds to a mutation probability of 1/6. The window size is then increased to $w = 60$ and 100, which corresponds to 3 months and 5 months respectively. After analysing the network at one time frame, the window moves forward in time by 1 time unit. Therefore, the time difference between consecutive moving windows is 1 day, and this allows the evolution of the network topology

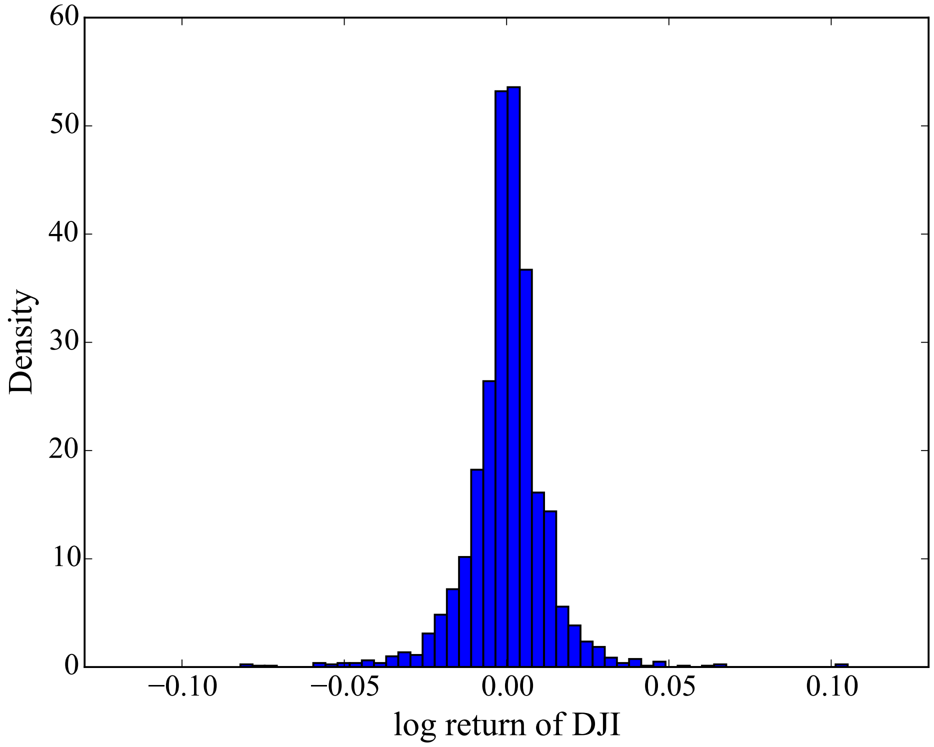


FIG. 4: Histogram of log return of Dow Jones Index (DJI) obtained during the period of mid-March in 2008 to the beginning of October in 2016.

to be captured. The changes in topology can then be compared with the performance of the market, for which DJI can be used as an indicator to evaluate the performance of all the stocks in the constructed network at different time frames.

V. RESULTS AND DISCUSSION

Three variables, namely the largest eigenvalue λ_1, tree length L of the MST and maximum degree ratio ξ, are obtained from the network constructed with dTWGA and compared with the market performance evaluated with the log return of DJI, which is denoted as S_{dji}, at window size $w = 20$. For fair comparison taking into account the effect of averaging, the average of the log return $\langle S_{dji} \rangle$ over the data points taken in a moving window with size w is also plotted in Fig. 5 as an indicator of the average performance of the market.

From Fig. 5, the instances of large decrease of all the three variables λ_1, L and ξ corre-

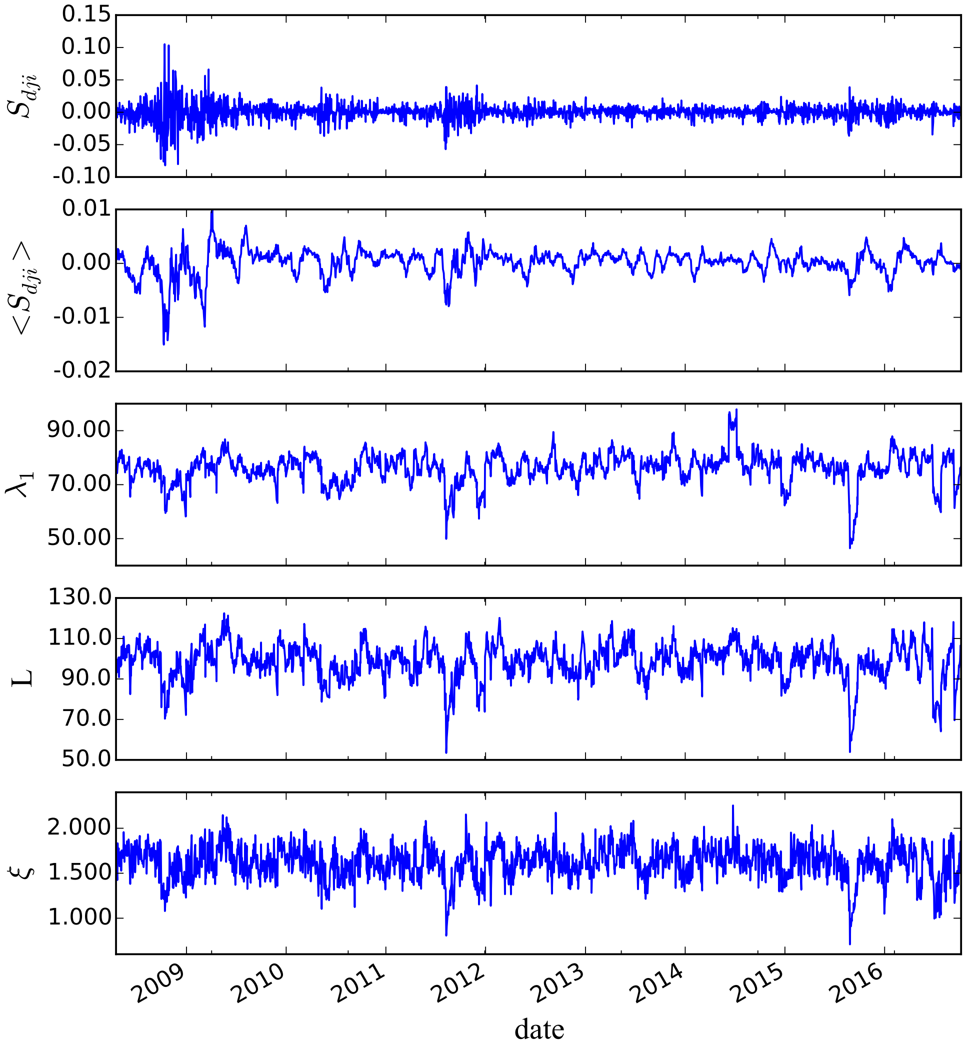


FIG. 5: Comparisons of the log return of Dow Jones index, denotes as S_{dji}, the average of log return in moving window S_{dji}, the largest eigenvalue λ_1 of adjacency matrix $\mathbf{D}$, the tree length L of the minimum spanning tree D_{min} and the maximum degree ratio $\xi(D_{min})$ for window size $w = 20$ as time proceeds.

spond to the large price fluctuations in the year of 2008, 2012, 2015 and 2016 as shown in the time series of S_{dji}. This correspondence becomes more apparent when comparing the trend of the three parameters with the average log return $\langle S_{dji} \rangle$. This suggests that the information of the overall environment of performance in the market can be reflected by information regarding the structure of the index component network. In particular, the tree length of the minimum spanning tree shrinks, indicating that the couplings between the stocks are

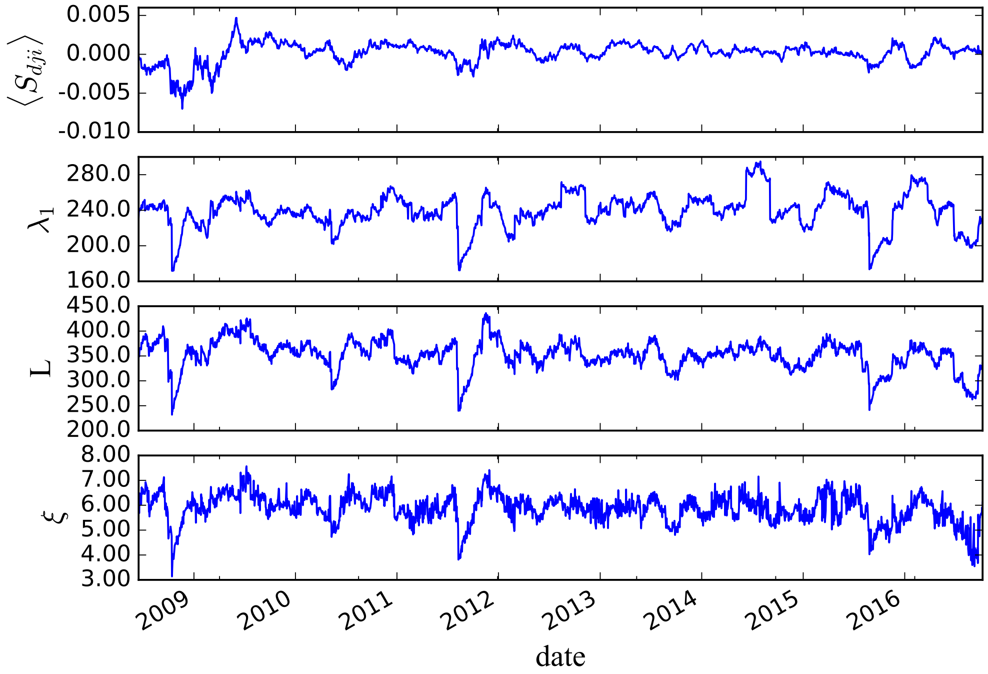


FIG. 6: Comparisons of the average of log return in moving window S_{dji}, the largest eigenvalue λ_1 of adjacency matrix $\mathbf{D}$, the tree length L of the minimum spanning tree D_{min} and the maximum degree ratio $\xi(D_{min})$ for window size $w = 60$ as time proceeds.

indeed stronger at the time of crash. The largest eigenvalue has a similar trend compared to that of the tree length of the MST. The maximum degree ratio also decreases at crashes, suggesting the effects asserted by the hub on its neighbours increases and therefore pulls the nodes closer towards it. The window size w is then increased to 60 and 100, corresponding to 3 months and 5 months respectively.

By comparing Fig. 5 with Fig. 6 and Fig. 7, the prominent features with large peaks and troughs are captured with both settings, of which the latter case with larger windows illustrates the remnant effects of the crashes on the topology of the constructed network over a long period. Another difference with larger window size w is that the effect of noise on signals is reduced since the signals are stronger. This effect is most apparent for maximum degree ratio among the three variables. While moving windows with larger sizes lead to more robust indicators against noise, the remnant effects would make it difficult to differentiate

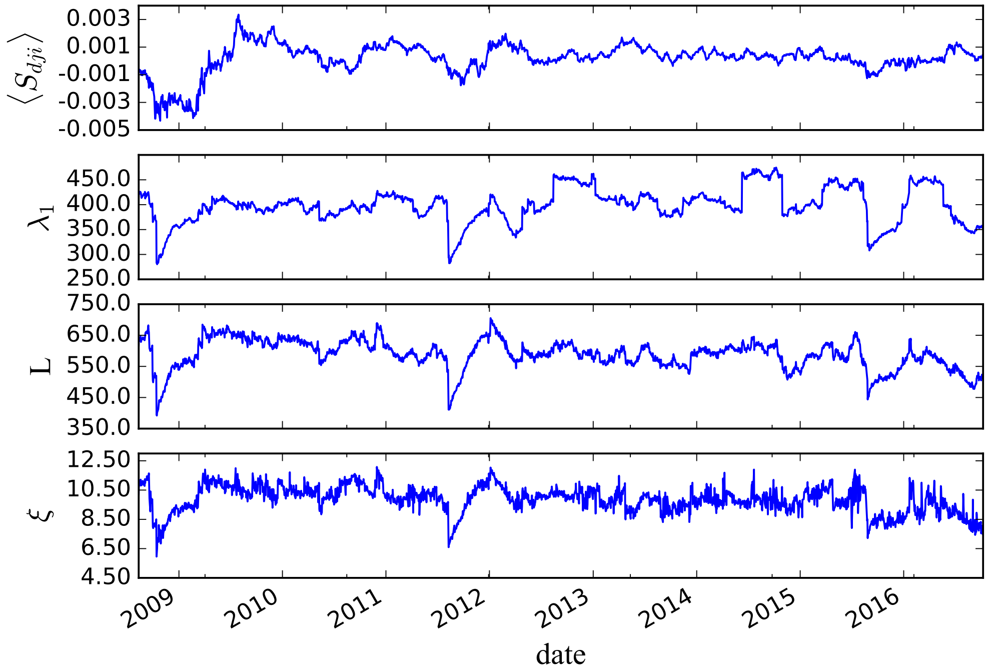


FIG. 7: Comparisons of the average of log return in moving window S_{dji}, the largest eigenvalue λ_1 of adjacency matrix $\mathbf{D}$, the tree length L of the minimum spanning tree D_{min} and the maximum degree ratio $\xi(D_{min})$ for window size $w = 100$ as time proceeds.

two large price fluctuates occurs in close proximity in time, and thus might not be as useful in predictions.

As demonstrated with the example of DJI, dTWGA can effectively describe the interactions between time series with few assumptions imposed. Further work will be conducted on applying the framework over different indices across different periods. As described in Sect. II, the entire methodology could be carried over to address unevenly spaced time series, such as those in high frequency trading. Our methodology would also be useful in the studies that merge different types of data collected in different ways, which could be sampled at different instances unlike the daily closing price studied here.

VI. CONCLUSION

In this paper, a novel algorithm dTWGA involving genetic algorithm is proposed. The algorithm is applied to recover the network structure of stocks and the evolution of the network over time is observed. It is found that the size of the network would shrink at large fluctuation of daily log return of stocks, showing increased dependence between the stocks, and the nodes would be pulled towards the node with the largest number of nearest neighbours.

[1] Yahoo. `https://finance.yahoo.com`

[2] Boss, M., Elsinger, H., Summer, M., Thurner, S.: Network topology of the interbank market. Quantitative Finance 4(6), 677–684 (Dec 2004)

[3] Cai, W., Chen, L., Ghanbarnejad, F., Grassberger, P.: Avalanche outbreaks emerging in cooperative contagions. Nature physics 11(11), 936–940 (2015)

[4] Han, R.Q., Xie, W.J., Xiong, X., Zhang, W., Zhou, W.X.: Market correlation structure changes around the Great Crash. arXiv preprint arXiv:1602.00125 (2016)

[5] Laloux, L., Cizeau, P., Bouchaud, J.P., Potters, M.: Noise dressing of financial correlation matrices. Physical review letters 83(7), 1467 (1999)

[6] Lin, Y.K., Yeh, C.T.: Maximal network reliability with optimal transmission line assignment for stochastic electric power networks via genetic algorithms. Applied Soft Computing 11(2), 2714–2724 (Mar 2011)

[7] Lui, G.C., Wu, D., Cheung, K.W., Ma, H.F., Szeto, K.Y.: Time warping of apneic ECG signals using genetic algorithm. In: Evolutionary Computation (CEC), 2016 IEEE Congress on. pp. 178–184. IEEE (2016)

[8] Luk, P.W.H., Lui, G.C., Szeto, K.Y.: Optimization of systemic stability of directed network using genetic algorithm. In: Computer and Information Science (ICIS), 2016 IEEE/ACIS 15th International Conference on. pp. 1–6. IEEE (2016)

[9] Mantegna, R.N.: Hierarchical structure in financial markets. The European Physical Journal B-Condensed Matter and Complex Systems 11(1), 193–197 (1999)

[10] May, R.M.: Will a large complex system be stable? Nature 238, 413–414 (1972)

[11] Metcalfe, T.S., Charbonneau, P.: Stellar structure modeling using a parallel genetic algorithm for objective global optimization. Journal of Computational Physics 185(1), 176–193 (Feb 2003)

[12] Onnela, J.P., Chakraborti, A., Kaski, K., KertÃľsz, J.: Dynamic asset trees and Black Monday. Physica A: Statistical Mechanics and its Applications 324(1-2), 247–252 (Jun 2003)

[13] Sefiane, S., Benbouziane, M.: Portfolio selection using genetic algorithm. Journal of Applied Finance and Banking 2(4), 143 (2012)

[14] Szeto, K.Y., Zhang, J.: Adaptive Genetic Algorithm and Quasi-parallel Genetic Algorithm: Application to Knapsack Problem. In: Large-Scale Scientific Computing. pp. 189–196. Springer, Berlin, Heidelberg (Jun 2005)

[15] Tse, C.K., Liu, J., Lau, F.C.: A network perspective of the stock market. Journal of Empirical Finance 17(4), 659–667 (Sep 2010)

[16] Tsinaslanidis, P., Alexandridis, A., Zapranis, A., Livanis, E.: Dynamic Time Warping as a Similarity Measure: Applications in Finance. In: 13th Annual Conference of Hellenic Finance and Accounting Association (HFAA) (2014)

[17] Ver Steeg, G., Galstyan, A.: Information transfer in social media. In: Proceedings of the 21st international conference on World Wide Web. pp. 509–518. ACM (2012)

[18] Wu, D., Szeto, K.Y.: Applications of genetic algorithm on optimal sequence for Parrondo games. In: 6th ECTA 2014-Proceedings of the International Conference on Evolutionary Computation Theory and Applications, Part of the 6th International Joint Conference on Computational Intelligence, IJCCI 2014, Rome, Italy. p. 30 (2014)